

CALL MSX
Número 8 • Junio 2008
Artículo
La Memoria en el MSX
Mapa fotográfico
Caos Begins
¡Vuelve la magia al MSX!
Majikazo

NÚMERO 8

JUNIO 2008



Sumario

secciones

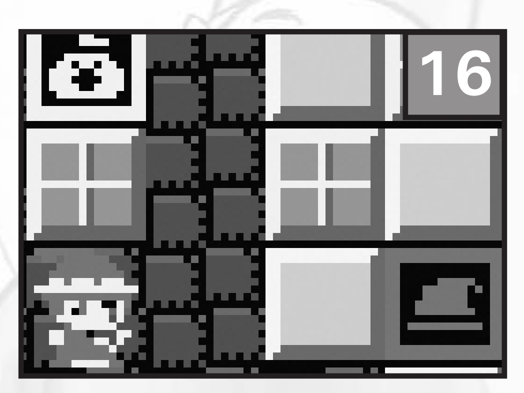
Reportajes: 30a Reunión de MSX de Barcelona

Otra gran Reunión para el recuerdo. Mucha presencia y novedades sorprendentes. Se vieron próximos proyectos de alta calidad.



Ya disponible: Majikazo

El grupo Lemonize nos presentó este vistoso juego que no podemos dejar pasar por alto. Gran título para nuestro MSX1.





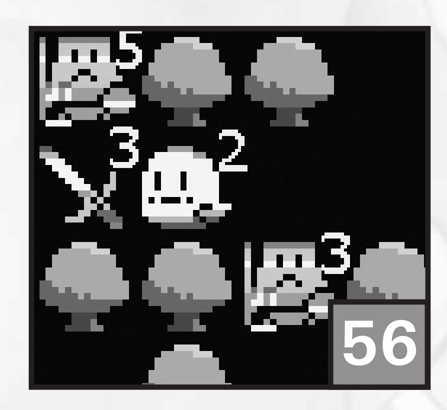
Mapa Fotográfico: Caos Begins

El ganador del pasado MSXDev al descubierto. ¡Ya no tendrás excusas para terminarlo!



Soft Amateur: Utopia

Lo que se puede llegar a hacer con un simple MSX1 lo vemos en esta excelente demo. Todo un lujo para los sentidos.



Cómo pasar: Griel's Quest for the Sangraal Excellent

Si estás atascado en este juego no te preocupes. Están todos los movimientos precisos para finalizar con éxito este cartucho.

Redacción

Francisco Álvarez
Roberto Álvarez
Óscar Centelles

Ilustraciones

Roberto Álvarez

Maquetación

Francisco Álvarez
Roberto Álvarez
Óscar Centelles

Colaboran

David Fernández
Avelino Herrera
Fernando López
Óscar Mate
Juanmi Ortuño
Manuel Pazos
Armando Pérez
Jesús Pérez Rosales
Antonio Robledo
Ramón Serna

Contacto

callmsx@gmail.com

http://callmsx.atlantes.com
por Avelino Herrera

EDITORIAL

REAPARICIÓN

Todo parecía indicar que ya no volvería a salir a la luz ningún ejemplar más de esta revista debido a asuntos personales de los redactores que nos impedían volcarnos en el proyecto como desearíamos. Pero supongo que la ilusión ha sido esa fuerza que nos ha llevado hacia delante y, aunque tarde, tenéis en vuestras manos un nuevo número de Call MSX.

Así que os daréis cuenta en seguida que no todos los comentarios son de ultimísima actualidad. Hay reportajes que quedan desfasados en el tiempo pero aún así creíamos conveniente publicarlos para que no hubiera una laguna temporal entre los dos últimos números. De este modo lo que váis a encontrar son los acontecimientos del último año y medio.

Si años atrás parecía que entrábamos en una tendencia de resurgimiento, en este último año hemos podido comprobar que esta tendencia se mantiene. En cuanto a soft hemos sido unos privilegiados, han salido juegos de primer orden con una calidad impresionante. El hard siempre está a la última, lo podemos comprobar con las nuevas tarjetas de Padial o con el MP3.

El tema candente sigue siendo el One Chip europeo. Los que no hemos podido esperar nos hemos hecho con uno japonés y realmente es una pequeña maravilla. Mejorando las actualizaciones se irán puliendo los problemillas que aún podemos encontrar en la compatibilidad. Tiempo al tiempo.

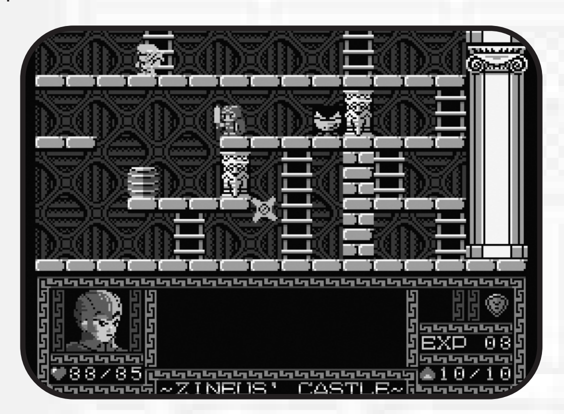
Y por último agradecer de verdad a los colaboradores voluntarios. Esta vez os lo habéis currado. Estamos muy contentos de haber recibido sin petición alguna artículos y trucos. Ese es nuestro objetivo, crear una revista para todos los usuarios, donde cada uno aporte lo que pueda y crear así unos contenidos más plurales y no limitados por los escasos recursos de los redactores. Mil gracias.

El equipo de Call MSX.

Nuevo éxito de MSXdev'07

Nadie esperaba que la edición de este año fuera tan brillante como lo ha sido, estando a la altura de los dos años anteriores; y es que hasta los últimos días del año no aparecieron los mejores juegos.

De las doce entradas en total, quien se ha llevado el gato al agua por méritos propios ha sido el juego **Caos Begins** del grupo *Hikaru Games*, quienes se han estrenado en este mundillo con esta pequeña joya de 32k. Con una sólida base inspirada en Maze of Galious rebosa calidad se mire por donde se mire y no ha dado opción a otros títulos interesantes que se presentaban al concurso.



El segundo premio es compartido por **Be Tiled!** de *CEZ Game Studio*, un divertido puzzle muy adictivo con sonido SCC y el original **Txupinazo** de *Imanok*, juego que nos traslada a las fiestas pamplonicas del mes de julio. Un juego realmente divertido y con una banda sonora memorable utilizando los chips de sonido PSG y FM.

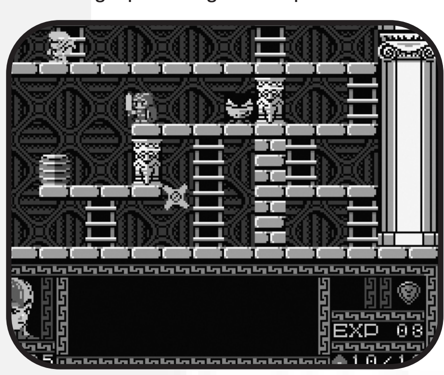
El tercer premio se lo ha llevado **Lotus F3**. Hacía mucho tiempo que no veíamos un título de conducción en nuestro sistema. *Daniel Vik* nos tiene acostumbrados a tecnicidades sorprendentes en el MSX y éste no es una excepción. Si hubiera hecho más hincapié en la jugabilidad estaríamos posiblemente ante el título que le hubiera hecho sombra al vencedor.

En resumen, MSXdev continúa siendo una buena plataforma para lanzamientos de software. Hay grupos asentados que repiten y otros nuevos que nos sorprenden. No nos podemos quejar, tenemos varios juegos nuevos con los cuales disfrutar una temporadita que no es poco. Estaría bien que los mejores juegos pudieran recopilarse en un cartucho, pero esto es sólo una idea.

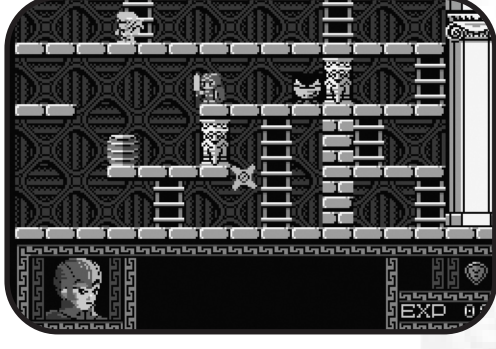
Vuelve One Eyed Willy...

Kralizec vuelven más fuertes que nunca. Tras **Majikazo** parecía que se daban una tregua. Sólo la aparición del famoso driver de la nueva LPE-MMC de Padial ponía de manifiesto que seguían en el candelero.

Y la noticia de un nuevo juego, llevado en el más absoluto de los secretos, ha vuelto a demostrar que son sin duda el grupo a seguir. La presentación ha sido esta vez a través de internet en forma de video. Un juego de plataformas, **The Goonies**, que presagia lo que el original de Konami pudo y nunca llegó a ser. No olvidemos que el Goonies original ya era genial, pero por lo visto en esta entrega, es mucho más fidedigno a lo que fue la película.



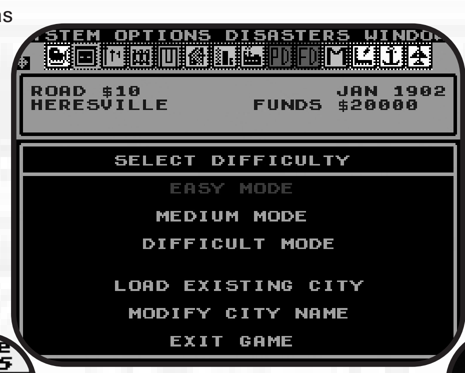
El paso a MSX2 y su screen 5, las músicas SCC a cargo de Carlo, el uso del v9958 si se tiene, 7 personajes con sus características propias, 4Mb de juego en formato cartucho... todo esto encierra un juego que promete ser grande y muy trabajado. ¡Una adaptación memorable del clásico de Richard Donner!



Y por lo que se rumorea no acaban ahí las sorpresas... así que habrá que esperar buenas nuevas o a tener el juego en nuestras manos, cosa que esperemos no se demore demasiado, pues parece estar bastante avanzado. ¡Da gusto dar noticias de juegazos como éste! ¡Enhorabuena y suerte, Kralizec!

Brasil I - Amusement Factory

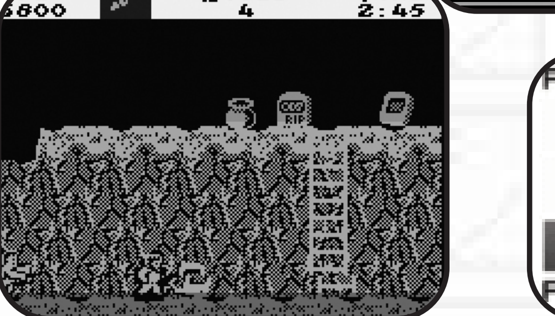
En los últimos meses el grupo brasileño *Amusement Factory*, viejos conocidos por el patch **Knightmare Gold**, nos está sorprendiendo con la conversión al MSX de varios juegos editados en su momento para los ZX Spectrum. Dado que en aquella época las compañías se olvidaron de nuestro MSX, ahora este grupo brasileño ha decidido llenar ese vacío. Los últimos juegos en versionar han sido **SimCity**, el cual en su día fue cancelada una versión específica para turbo R, y el mítico **Ghost 'n Goblins** de la empresa Capcom con parche para evitar bugs incluido.

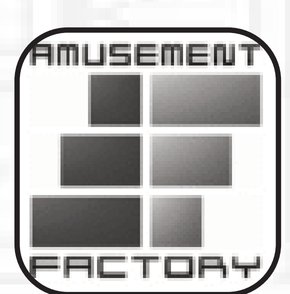


El resultado es más que correcto, pero si disponemos de un MSX que supere los 7 mhz, es decir un MSX2 con z80H o un turbo R, disfrutaremos de mayor suavidad en el control.

Además de este juego también han hecho conversiones de otros clásicos como lo son el **Shinobi** y el **Altered Beast** de Sega, con las mismas características técnicas que los anteriores.

Sin duda son juegos que nunca hemos podido disfrutar y, más vale tarde que nunca, el MSX dispone de ellos.

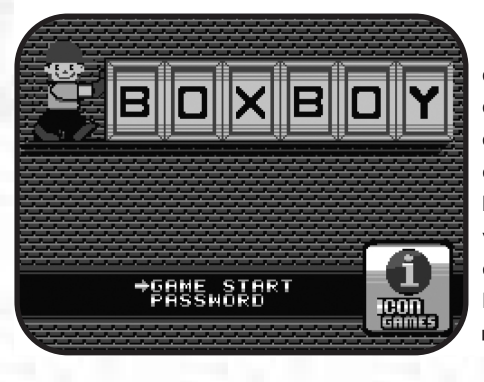






Brasil II - Icon Games

El programador brasileño conocido como *Slotman*, creador de la versión en cartucho **Jungle Hunt** de la consola ColecoVision está desarrollando la conversión de **Montezuma's Revenge**, juego de la misma consola y que promete mucha jugabilidad. Esperamos tenerlo pronto y en versión cartucho si es posible. No contento con esto último, ha mostrado dos novedades, aunque una de ellas también está aún por terminar.



Para la pasada MSX'Rio 08 dejó la aparición de su nueva conversión: **BOXBOY**!. Una versión del clásico Sokoban para MSX1 con 6 mundos y 150 pantallas repartidas entre ellos. Aprovechando que Lotus F3 innovó en la publicación de los High Scores por internet, *Slotman* ha querido aplicar la idea a BOXBOY también. Una lástima que no pudiera pasarse por allí para mostrarlo y así tener referencias de los usuarios sobre el juego.

Su última aportación a las novedades, inconclusa hasta el momento, es el **Transylvania 3 - Vanquish the Night**, una nueva conversión. Esta vez se trata de una aventura gráfica clásica o AVG, muy parecido en su forma de interactuar con el usuario al mítico Perry Mason.



Nijmegen 2007



Han pasado ya unas dos semanas desde mi tercer viaje a Holanda. Para unos la tierra del sexo, para otros la tierra de las drogas. Mi realidad dice que es la tierra del MSX (y de las ovejas). El paisaje bucólico que uno ve de tren en tren al recorrer el país está siempre adornado de grandes ovejas que dan un queso... ¡hmmm!

Bueno, basta de que se me vaya la pinza y vayamos al grano. En Holanda, aparte de visitar a muchos MSXeros, tuve la oportunidad de ir a Nijmegen '07, la feria del MSX más importante en Holanda probablemente tras Bussum, en octubre, y sin duda alguna la que centra toda la atención del sector en los seis primeros meses del año. Allí se cocían muchas cosas, pero se podía respirar al MSX, amigos, por todas partes.

No estaría mal que os hiciese una foto con palabras de los puestos que por allí había. Seguramente olvidaré algunos, también a gente; que me perdonen ellos, pero seguro que de lo importante se me escapa poco. Allí, por ejemplo, estaban los miembros de *MSX Posse* teniendo una charla continua, que duró todo el día, así como los *Lammertsma* a su lado. Este clan, formado por *Age* y por *Sjoerd*, es sin duda uno de los más frikis que he conocido jamás. Como buenos MSXeros, organizaron un concurso que tuvo rodando la pelotita del Roller Ball todo el día. Sí señor, adictivo ese juego como él sólo. También había otros puestos adyacentes con ordenadores del año de la pana, Colecovision y esas cosas... ¡Me hicieron sentir algo viejo! *Bas*, el hombre que repara MSX allá donde va, se trajo una pila de MSX (en verdad eran tres pilas, unos doce ordenadores, uno puesto sobre el otro) y en uno de ellos no hizo sino conectar Metal Gear 2: Solid Snake. ¿Os suena el juego? Seguramente no, al fin y al cabo es un juego más, ¿eh?

Seguid recorriendo puestos. Vais a ver el de Sunrise, perfectamente preparado para las ventas, como siempre, allá donde va. Uno de sus miembros, *Koen*, suelda y prepara hardware al instante, como siempre. Otro se dedica a dormir (según me cuentan, anduvieron preparando todo la noche





anterior) y el tercero, *Rob Hiep*, es el hombre de las ventas. Yo me pillé un Akin. *Cas Cremers* se merece eso, ¡qué menos!

Más. Había un puesto de venta retro con cosas para la N64, Megadrive y Dreamcast. A mí siempre me agrada ver cosas diferentes en las ferias del MSX, aparte de los MSXs en sí. Al fin y al cabo, me gustan los videojuegos en general. Y le añaden un colorido especial a una feria.

DeltaSoft es el siguiente puesto, señores, con sus conejos inconfundibles. Estuvieron exponiendo algunos de sus juegos como el famoso (¿es famoso o me lo parece a mí?) *Konami Quiz 2*, ¿se llama así el juego? Da igual, seguro que sabéis al que me refiero. Pero me dio la sensación de que venían cargados con otros juegos diferentes que fueron cambiando según iba avanzando el día.

[D-Tail] y *AuroraMSX* estaban en otro puesto por aquella zona. No diré lo que estaban haciendo porque es en este caso confidencial. Lo importante de estas reuniones, y quizás esto te interese más de lo que piensas, lector, es que en reuniones así es donde muchas veces se cuecen los comienzos de los proyectos. Júntate con dos o tres que sepas que van a dar el callo (si no, olvídate), habla de tus ideas y si estás por la labor de ahí sale un juego. O una pequeña colaboración, que puede parecer en un principio, y que termina convirtiéndose en un juego excelente (no complejo necesariamente) simplemente si todos mantienen la constancia y la ilusión de fabricar algo con la mente, y luego lo llevan a cabo en la realidad.

No podéis captar los sonidos que emana esta reunión, una atmósfera en la que creo que de todos modos ya estáis muy metidos. Imaginaos los sonidos del Roller Ball a toda pastilla,

Uno de los puntos fuertes del día fue el artilugio de *NoWind*, que ha hecho posible conectar el MSX a un PC vía USB. Es visualmente chocante, pero cada vez menos, afortunadamente: nos estamos acostumbrando a ver el MSX conectado a nuestro PC cada vez más. En mi opinión, esto es muy bueno, hace el MSX más universal. Lo próximo, algo con lo que la gente disfrutaría de verdad del MSX, a lo mejor hasta sin darse cuenta, sería patentar un sistema de descargas de juegos del MSX... exactamente lo que la Wii japonesa parece que va a hacer, pero sin restricciones estúpidas de número o tipo de juegos. En fin, algo es algo.

Y hablando de la Wii, se podría decir que fue el invitado especial de la jornada. Estuvo conectada y siempre rodeada de gente. Yo personalmente eché de menos algo del MSX ahí, y que conste que la Wii en sí promete una barbaridad. Lo importante de todos modos es que mucha gente disfrutó con ella.





la música de la intro del Metal Gear 2 de fondo, la gente gritando con la Wii, más otros muchos samples remezclados, el olor a café y... ¿ya está? Bien, podemos continuar.

Al lado del puesto de la Wii estaba el del *OpenMSX*, ese emulador cuyos creadores dicen que aquél es el más ambicioso de todos. Muchos, incluido yo, usan el BlueMSX, pero el esfuerzo de este equipo por mejorar el suyo es encomiable.



Además, ¿dónde funciona el Monster Hunter aparte de en un MSX real?

En frente tenemos a *Wolf*, que para mí ya es una leyenda viva del MSX. Es un tío que no habla mucho en las ferias porque no para de hacer cosas incluso en ellas, especialmente en lo concerniente a remakes de canciones del MSX, asunto en el que es un maestro. Me sorprendió, como más tarde indicaré, que no contento con ello, también se dedica a planificar videojuegos, a dibujar gráficos y a programar editores de niveles y de scripts que posterioremente se aplican en la fase de desarrollo de un juego. En fin, lo más curioso de todo es que apenas hablé con él en la feria. Fue en otra ocasión, en la que fui invitado a su estudio para ver cómo hacía música, en la que tuve la oportunidad de meter mi hocico (buga buga) en tal oficio.

A su lado estaba *Podatron*, un usuario alemán que también enseñó sus cositas en la feria, tal cual Symbos, en el que tenía ejecutándose a la vez Pac-Man, Matrix y alguna otra cosita.



Para el final dejo a *Poke*, que también estaba en esta fila de "desarrollo", y al cual fue un placer conocer en carne y hueso tras seguirle varios años musicalmente hablando cada vez que publica, por ejemplo, uno de esos Airwaves que tan chulos y retros son. A mi juicio, aparte de ser un tipo amable y con los pies en el suelo, me pareció un artista, creando con su Amiga vídeos que contenían gráficos de antaño y música parecida a los conocidos como 'chiptunes'. Algo parecido a los videoclips de los 80 y que yo llamaría 'Retroclips', porque son auténticamente Retro y creo que merecen un nombre genérico que indique la calidad de su producción.

Aparte de los puestos, merece la pena mencionar algo de lo que pasó en la feria, ¿no? Por ejemplo, comentando la gente que fue al evento, localizado en un centro público que viene a ser de grande como lo que fue Oss en su día, pero con más iluminación sin duda.

Se acercaron a la feria, aparte de los ya comentados, *Lord*



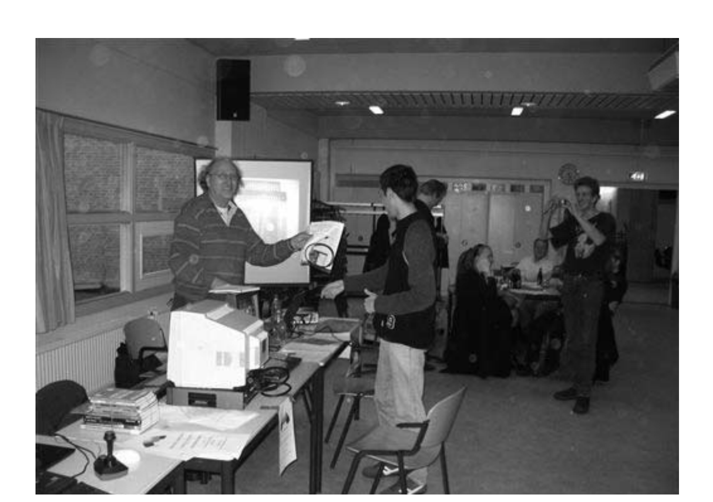
Robert y Remco, que llegaron algo más tarde pero que hicieron acto de presencia. Ro también se acercó amablemente a saludarme, al igual que el mencionado AuroraMSX. Fixato fue otro de los asistentes al evento. Y que me perdone quien no esté en esta desordenada lista...

Por supuesto, mención especial merece Quibus, el organizador de la feria y al que se le debe mucho con respecto a estas reuniones MSXeras del norte de Europa. ¡A él lo coloco al final pero debería ir al principio!

Conclusión: El evento fue un éxito. La razón es simple: lo pasamos bien, disfrutamos del MSX y, por qué no, de lo que no fue MSX: el cafetito de la mañana, la comida del mediodía o la genial cena en el chino en el que los niños recogían lo que dejaban los mayores. Shhh, ¡no digáis nada a inmigración que la comida estaba de rechupete!

Zett, también su novia, *Marinke*, la chica rubia que siempre luce tan buen tipo de MSX y que da un colorido especial a los eventos del MSX. *Zett* y yo tuvimos una charla interesante acerca de Bezz, trató de explicarme (casi de coaccionarme amablemente en mi posición de miembro del jurado de la MSXDEV 06) por qué su juego merecía una buena puntuación en dicha competición. Hice bien al fijar mi veredicto antes de ir a Holanda para evitar cualquier tipo de interferencia con mi propio juicio, que puede ser subjetivo pero que siempre trató de ser justo y de valorar el esfuerzo desinteresado de cada desarrollador. De esa manera disfruté de todas las conversaciones sobre el concurso, ya que al fin y al cabo el trabajo estaba hecho... ¡Qué complicado fue callarse los resultados, pardiez!

Estuvieron por allí *Bas*, el eterno reparador de ordenadores, *Jorito* y *Arjan*, amigos del *Team Bomba*, así como



La experiencia

El hecho de viajar a un país extranjero durante diez días, visitando a gente del MSX aquí y allá, conociendo cosas nuevas y otras no tanto es algo que merece la pena comentar. Al final lo que queda no es sólo la reunión del MSX, de hecho eso suele ser lo más formalito. El visitar y estar en casa de los amigos del *Team Bomba* me propicia horas y horas de MSX, ya sea de juego, de escuchar canciones o sencillamente de tener una conversación interesante sobre nuestra preciada afición. También me di cuenta en esta ocasión de los beneficios de ser un MSXero de pro. *Wolf*, por ejemplo, ya ha compuesto bandas sonoras en películas oficiales y en series de nivel nacional tras haber aprendido mucho con el Moonsound y con el MSX.

Y por lo tanto...

Os recomiendo disfrutar con pasión de esto que es el MSX. Si no podéis asistir a las ferias no os preocupéis, es lo de menos, y si se planea bien llega uno u otro año. Siempre se puede organizar una reunión local o un "vicio" con los compañeros.

Larga vida al MSX.

Jesús Pérez Rosales aka [DK]

30ª Reunión de Usuarios de MSX de Barcelona

Y es que las Reuniones de fin de año atraen últimamente a gran número de visitantes, llegando a cifras que se acercan a las míticas de hace una década. Mucho tiene que ver, nos tememos, que es una fecha en que suelen acabarse muchos proyectos y se presentan en sociedad en dicha cita. También el calendario, si hay días de puente se pueden acercar visitantes de fuera de Barcelona.

Lo primero que nos sorprendió fue el número de expositores, en este caso nueve. Así que hubo una sensación de lleno absoluto, debido al tamaño de la Sala, que no es tan grande como en las añejas reuniones.

Como viene siendo habitual el primer stand que nos encontramos es el de **Segunda Mano** a cargo de Jordi Tor. Ya se sabe, usuarios que se "quitan" material de encima y otros ansiosos buscadores de joyas que a veces encuentran el artículo deseado a precios no siempre asequibles pero, si lo quieres, lo pagas. Parece ser que estuvo bastante bien de género, algún juego raro de la saga Ultima, inspirada en épocas prediluvianas, entre otros títulos y vimos también in situ algún periférico exótico como una tableta gráfica Philips o un MidiSaurus de Bit2. Toda una oportunidad para los coleccionistas de turno.



Allí estábamos nosotros, **Call MSX**, presentando el séptimo ejemplar de este fanzine que ya va poco a poco haciéndose un lugar dentro de la familia MSX. También estaban disponibles varios de los números atrasados. Como siempre llevamos un TurboR bien equipado para mostrar cosas. En este caso los juegos que aparecen en la sección Software Amateur, que llamaron bastante la atención del público. También gracias al señor Turón pudimos probar el genial adaptador para mandos de PSX creado por Tsujikawa, que además hace juego con el color de la carcasa del OCM japonés.





30a Ru de Barna

Los chicos de **Moai-Tech** que habían sacado el número 4 de su webzine nos sorprendieron con la venta de figurillas de arcilla de personajes MSX y CDs con archivos *PDF* de su revista de papel. También les quedaban antiguos DVDs recopilatorios de música de MSX. No olvidaron traerse un TurboR para que la gente se echara unas partidas.

Seguimos con los compañeros de **Paxanga Soft** que aunque no tenían nada nuevo que vender sí enseñaron lo que estaban haciendo en esos momentos. El primero de ellos un juego plataformero para MSX1 "*Tam Tam Twins*" con scroll e inspirado en el Pc Kid, aunque tiene toques de Mario y Sonic. Muy espectacular visualmente gracias al trabajo de Rc-743 y donde parece que colabora musicalmente Manuel Dopico, que regresa a la escena después de varios años de inactividad. El otro proyecto no sabemos el nombre que tendrá pero es tipo aventuras ambientado en cavernas.



Otros que vuelven a ser ya habituales son los **MATRA**, con un stand a rebosar de productos. Camisetas cachondas con temas relacionados con el frikismo televisivo y los videojuegos que daban mucho color. Como novedad dos cartuchos de bella factura: *Saimazoom* y *Griel's Quest for the Sangraal* al precio de 10 euros (más ajustado imposible). Y como siempre te puedes hacer con copias de los antiguos discos de Matra y Kralizec o del primer cartucho *Ink*.



Otro grupo que pese a no traer nada nuevo da mucho ambiente a la Reunión en forma de música a toda pastilla con los MSX conectados a un amplificador. Sí, son los **Taburoto**, un grupo que prefiere pasárselo de coña teniendo stand propio y disfrutar el día con su ordenador que aparecer como simples visitantes. Digno de elogio.



Los **Karoshi Corp** son un grupo de currantes y nos presentaron los proyectos en los que están metidos que no son pocos. A destacar la demo no jugable todavía del *Speed Jet Racers*. También dieron a conocer juegos que estaban en la fase final del concurso MSXdev como el *Monster Hunter* de Nerlaska o el *Beepertron* de Dioniso. Y por último vendían unas chapas graciosas con sus propios personajes como Malaika.





El hombre del norte también bajó a Barcelona y montó su propio stand. **Manuel Pazos** consiguió que la gente se animara a venir a la Reunión para comprar sus productos. Sobretodo el esperadísimo cartucho *MultiRom*. Un cartucho que como bien sabéis incorpora juegos de anteriores MSX-Dev y alguno nuevo como Majikazo, además de las versiones mejoradas de Operation Wolf y Traffic Jam. Todo un pedazo de cartucho con la gracia de cambiar de juego cada vez que reseteamos. También vendía otro cartucho recopilatorio con juego míticos de MSX1 llamado "*Children Collection*". Se trajo uno de los pocos prototipos disponibles de MSX One Chip (una versión anterior a la oficial japonesa, sin carcasa) y presentó una joya de del genio Tsujikawa, el Mega SD. Todas las ventajas de un Mega SCSI en un pequeño cartucho con tarjeta SD para almacenamiento. Una pequeña maravilla, pero debido a que está implementada en el OCM parece que no fabricará más.

Y por último el stand más surrealista de todos los tiempos, **One Beer MSX**. Makinavaja, conocido en esta revista como la "estrella" de la Reunión, el usuario más cervecero, creó escuela y montó un stand con su colega Xenon para que la gente echara una partida en su ordenador mientras se bebía un vaso de birra. Dos vicios en uno, ¿se puede pedir más?

Como suele ser tradicional hubo cena multitudinaria para todo el que se quisiera apuntar y acabar con una jornada completa de MSX.





31ª Reunión de Usuarios de MSX

Barcelona

Siendo éste uno de los encuentros más flojos y sabiendo que puede ser un poco pesado comentar tres reuniones en un mismo número haremos un repaso muy breve de lo que fue esta cita del 28 de Abril de 2007.

Una Reunión que se propuso tarde y mal, con más bien pocas novedades dado el poco margen con la edición pasada pero que aún así logró que la asistencia no fuera ridícula. Al final hubo más ambiente de lo que se esperaba y salimos con un buen sabor de boca.

Para empezar el número de stands bajó a seis y eso que se apuntaron a última hora. Y de éstos tres no traían novedades.





Como siempre el género del stand de **Segunda Mano** siempre va variando, faltaría más, y se llegó a ver alguna rareza. Los que disfrutamos con las ilustraciones de los juegos es una ocasión genial para ver portadas de juegos raros, y algunas de mucha calidad. Quizá el juego de más renombre que se expuso fue el Burai pero había títulos antiguos que no habíamos visto nunca el package con anterioridad.

Call MSX no publicó ningún número nuevo pero remaquetó el primero que ya estaba agotado y le dio un look acorde con los tiempos actuales. El número 2 también se reeditó y nos sorprendimos con la nueva calidad en la impresión, con mayor definición. En nuestro stand se presentó el primer One Chip MSX japonés propiedad de David Turón, el hombre que lo tiene todo. Nos dejó impresionados este nuevo chisme, ¡es tan pequeño y cuco! Se probó el Moonsound y funciona de maravilla.

MATRA por su parte trajo más diseños de camisetas y nuevos cartuchos como el *Sudoku*, *Beepertron*, *Phantomas Infinity*, *Magical Stones* y una recopilación de juegos de Konami.

El último stand que trajo novedades fue **La Sekta**, quienes fabricaron a mano unos pedazo de joysticks tipo Telemach que dejaron probar a los interesados en hacerse con ellos. También sacaron un adaptador para mandos de Megadrive y así jugar con ellos en el MSX.



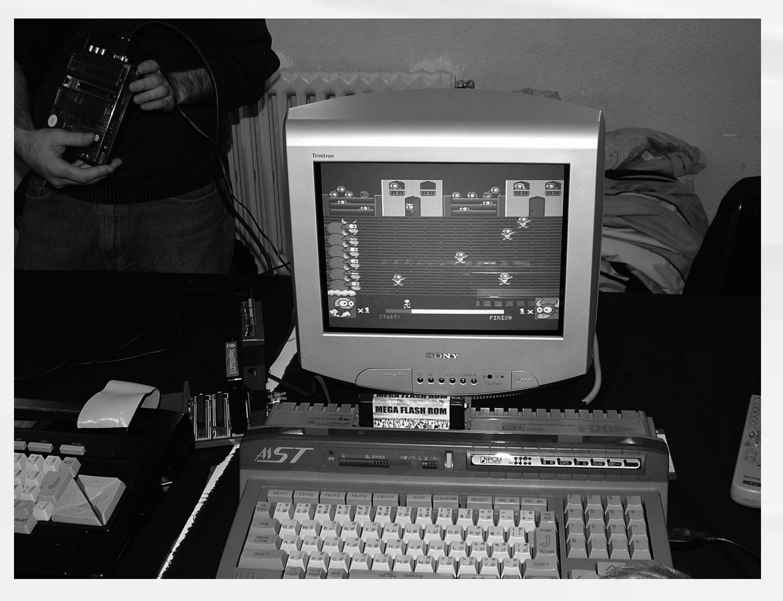
32ª Reunión de Usuarios de MSX
Barcelona

Sí amigos, no hay dos sin tres. Pudiera parecer que este evento tendría que ser comentado en el siguiente número, pero como no ha sido posible y no hemos querido dejarlo pasar por alto, terminaremos una trilogía de eventos Barceloneses en las dos próximas páginas.

A finales de diciembre de 2007 se produjo la 32ª Reunión de Usuarios de MSX. Tras más de 15 años y con un encuentro anterior bastante flojo en cuanto a novedades y número de asistentes se refiere, nos adentramos de nuevo en una senda positiva que augura tiempos futuros mejores.

Nuevamente el número de Stands fue inferior a la 30ª edición, aunque los contenidos esta vez no defraudaron. Si bien es cierto que el equipo de **call MSX** no presentamos ejemplar nuevo alguno (y esto era tradición desde hace 4 años) al menos sí montamos stand, con nuestro propio OCM conectado a un 1084 de Commodore que se adquirió en el mismo encuentro. La conexión mediante cable VGA-RGB, gentileza de David Turón, atrajo el interés de algunos visitantes. Y aunque ya estaba preparado para la ocasión, esta vez sí podréis disfrutarlo en este mismo número: cómo hacer dicho cable, por **Manuel Pazos**.

Y sí, el susodicho volvió a aparecer por Barcelona, compartiendo stand como un año atrás con Kralizec. Esta vez se traía unas nuevas MegaFlashROM SCC construidas desde cero, carcasa incluida. Mismas carcasas que desde entonces han disfrutado otras producciones como Manbow2 o Caos Begins.





¿Hemos nombrado a **Kralizec**? Pues sí, los auténticos emblemas del evento. Dos de sus miembros se dejaron caer: Carlo Bandini y Armando Pérez, si bien sólo este último estuvo en el stand. Nos dedicó unas palabras para presentar su última creación, un driver que por entonces estaba en versión beta de la tarjeta LPSDMMC de Leonardo Padial y un prototipo de esta última.

Y el driver encandiló a más de uno, pues consigue que por fin tengamos disponible un sistema de almacenamiento con soporte FAT16 y cargador de DSKs sin necesidad de tener un MegaSCSI. Es más, es compatible con MSX1 y con la última versión del DOS2. ¿Qué más puede pedirse? Pues las bienvenidas explicaciones por parte de Armando, que disipó cuantas dudas e inquietudes relacionadas con el producto que mostró el personal. Está claro que un buen número de ventas del mismo se fraguó aquí. Como si esto no fuera poco, Imanok nos sorprendió en este mismo stand con la presentación de su nuevo juego que tenía intención de presentar en la MSXdev'08. Calurosa fue la acogida que recibió a Txupinazo por parte del respetable.

También **Moai-Tech** dejaron verse para la ocasión y para no perder la costumbre nos acercaron *merchandising* MSX con el que conseguiremos tener una vivienda más MSXera si cabe: desde sus ya tradicionales minimuñecos hasta posavasos con algún que icono de nuestros juegos favoritos, como el personaje principal del Ninja Kun. Y cómo no, las pegatinas MSX y sus CDs. Entre el material pudimos encontrar lo que en un futuro será nuevo Hardware para nuestros MSX por parte de La SeKTa, ¡un adaptador de puerto ps2 a DB9 para poder conectar ratones de PC! Genial idea, esperamos que la lleven

a cabo pronto. Nos congratula tener un stand tan variado como éste, es un lujo encontrarles en cada Reunión, pues siempre te llevas algo nuevo para casa. Amenazaron también con realizar una nueva versión de su webzine, la quinta, en un tiempo razonable.

Y con ellos estaba **Julio "Jagur" Gracia**, que tuvo en su stand adyacente algunos juegos originales a la venta, varios difíciles de encontrar y otros bastante desconocidos para el usuario medio, haciendo sana competencia con las ventas del célebre y siempre concurrido stand de **2ª Mano** (o 2ª zarpa, como diría su expositor por antonomasia, Jordi Tor).

Dicho stand de 2ª mano tuvo cosas interesantes, como un MSX2 Mitsubishi MLG3, aunque su estado no fuese óptimo. También hicimos caso al lema de la caja "No lo toques, no despiertes la furia del dragón" y no nos llevamos el Double Dragon que allí podía verse.

No podían faltar el stand de **Karoshi Corp.** y no nos defraudaron. Inundaron el stand de cartuchos Karoshi aunque siempre acaban faltando. Y es algo de agradecer, que cuando preguntes por uno en particular la respuesta sea "ya se han agotado"; aunque sea por el bien del sistema y sus ventas. Allí podrían encontrarse las últimas novedades como el Malaika o el Sudoku entre otros, así como el Phantomas Saga Infinity.

No podemos olvidarnos de hacer una petición formal a la AAM y es la inclusión de una barra de bar para que Toni, más conocido como Maki, se tire unas cervezas en su **1BeerMSX**. Y es que siempre es agradable degustar una birra, ¡aunque sean Xibecas!



A destacar el torneo del Greatest Driver 2D Special de T&E Soft que aglutinó a veteranos y noveles (!) bajo un mismo MSX y que, como cabía esperar, ganó Imanok.

Desde aquí agradecer a todo el personal que por allí se acercó, sobretodo de fuera de Barcelona: Daniel Zorita, Ramoni, Nerlaska, Konamiman, Sapphire, JLTursan, etc. ¡Esperamos veros en la próxima con un nuevo número!

Por último, dar la enhorabuena a los nuevos integrantes de la AAM y desearos suerte en la elaboración de la nueva y posteriores Reuniones de Usuarios de Barcelona. Esperemos que la baja para la siguiente de Imanok no reste calidad en la participación en los torneos y sí añada una competitividad más reñida entre los participantes.



Ya disponible



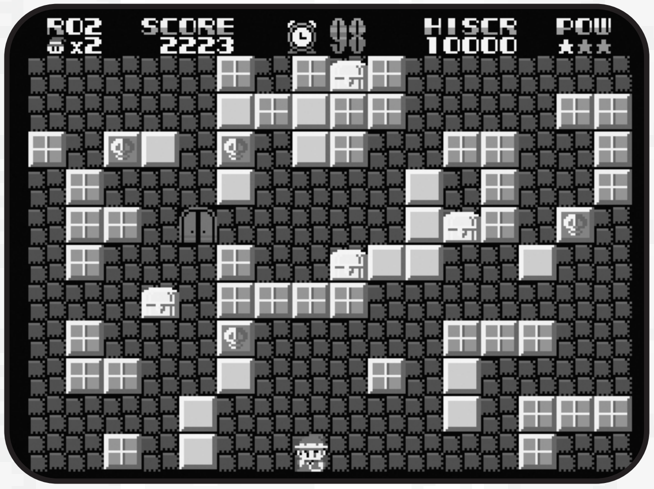
Majikazo

Magia en tu MSX

En el esperado cartucho Multirom de Manuel Pazos que se vendió en la 30ª Ru de Barcelona hubo una sorpresa más que grata, la inclusión de una nueva Rom que no esperábamos. Su nombre Majikazo y no parece un juego de relleno precisamente.

El autor de este juego es el grupo **Lemonize**, pseudónimo de **Kralizec** para juegos originales de MSX1. Un juego que podía haber apostado perfectamente por participar en el concurso MSXDev, de hecho su configuración entra dentro de los requisitos, y en cambio han optado por no hacerlo.

Majikazo es un juego de MSX1 y de 48k de memoria, con la capacidad de poder ejecutarlo a 60 hercios pulsando *Select*. Si disponemos de MSX2 o superior, al pulsar *Graph* al arrancarlo se emulará la paleta original del MSX1, sino será la de MSX2. Aconsejamos 100% seleccionar 60 hercios ya que la diferencia en velocidad es asombrosa, pareciéndonos a 50 hercios demasiado lento.

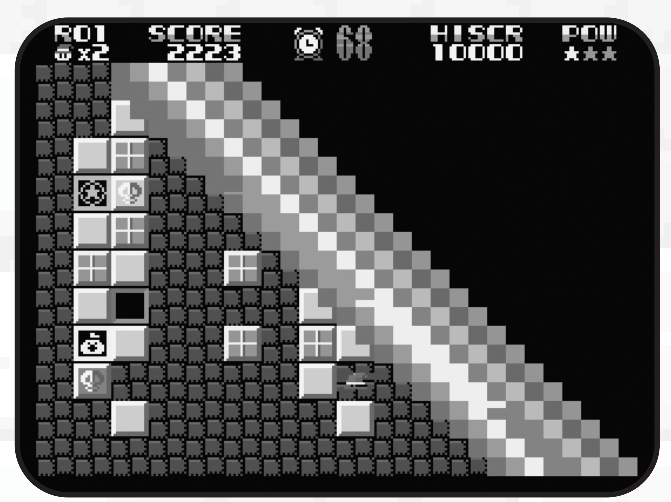


El apartado gráfico es definido y colorista

Majikazo es un clon del mítico juego de las recreativas *ToyPop* (Namco 1986). Lemonize ha adaptado el clásico y se lo ha apropiado cambiando la historia y los personajes pero manteniendo intacta su filosofía. Como en toda adaptación a MSX de un juego de arcade hay que ser conscientes de las limitaciones de nuestro ordenador personal y a primer golpe de vista parece que esta conversión ha salido airosa.

El aspecto visual es generoso, mucho color, y eso gusta. Además está puesto con gracia y los gráficos, pese a ser un juego de vista cenital, son divertidos, claros y nítidos. Todo está bien resuelto, hasta la tipografía. No lo vemos cojear en ningún aspecto, es muy profesional. Majikazo no pretende ser un juego realista como lo pudiera ser Metal Gear 2, por lo tanto se le perdona la carencia de perspectiva. Quizás hubiera sido un punto a su favor buscar un punto de perspectiva en los personajes, darles un "picado" para dar más sensación de altura, pero hubieran sido también más confusos los gráficos debido a la limitación cromática de los sprites. Así que los gráficos responden a las expectativas. La portada nos puede parecer un poco simplona, pero pocos juegos de MSX1 tienen una gran presentación; fijaos en Konami, donde casi siempre sólo aparece el logotipo del juego y nada más. Un trabajo más que correcto del grafista Sutchan, el cual vemos que se ha supe-rado respecto a otros trabajos anteriores. Los personajes son menos “toscos” (también más pequeños) que en Dahku o el inacabado Dream On y tienen mayor soltura en el movimiento. El protagonista también posee carisma, detalle que da más personalidad al global del juego. Sin duda, es una nueva evolución en la carrera de los aquí ocultos Kralizec.

La música es otro de los grandes aciertos de Lemonize. El “fichaje” del músico profesional Carlo Bandini se hace notar. Los temas son adecuados para el juego y la música se te quedará grabada en la memoria para siempre, sobretodo la de la presentación. Ha creado un sonido muy particular para este juego, teniendo los temas una sonoridad homogénea. Es de agradecer la utilización del MSX Music en un juego de MSX1, algo por otro lado no siempre habitual. Además, los efectos de sonido están en la onda de un juego de estas características. Son efectos correctos, breves pero ajustados, que no dañan la melodía, más bien la potencian. Aunque hubiéramos preferido más cantidad y variedad a los ya presentes, con el actual número estamos más que satisfechos.



Bello efecto de barrido de pantalla

Por último nos metemos de lleno en la jugabilidad. Ramones ha vuelto a cumplir. Ha conseguido crear un juego atractivo y divertido. Y encima, los movimientos de los sprites son suaves en general. Todo está muy ajustado y no encontramos ralentizaciones ni parpadeos. Además tiene detalles asombrosos para ser un juego de MSX1 como puede ser el lanzamiento de las llaves hasta la puerta desde cualquier punto de la pantalla. Lo más parecido a este efecto lo hemos visto en King's Valley cuando se lanza el cuchillo, pero en este caso siempre es en dirección este-oeste. En Majikazo, como vemos, se han superado en complejidad y además es acertada la animación de la rotación de la llave mientras viaja en diagonal. Otro aspecto a destacar es el momento de superar una fase en la que se borra la pantalla mediante una franja multicolor que la barre en diferentes direcciones. Un efecto digno de demos trasladado a un juego.

Si nos fijamos en todos los detalles nos daremos cuenta que se ha hecho un excelente trabajo. Desde la "inteligencia" de los enemigos (mira que nos llega a poner nervioso ese diablillo rojo que nos persigue y nos dispara) hasta la pérdida del gorro por parte del protagonista (que nos recuerda a la amputación de alas en *Twin Bee*). Magias espectaculares, como la que produce una onda expansiva, y magias apropiadas para eliminar cada enemigo concreto. También tiene objetos interesantes que pueden beneficiarnos o perjudicarnos. Es un juego más complejo de lo que a primera vista pueda parecer.

La dificultad estriba en deshacerte de los enemigos, ya que cada uno se mata con una magia adecuada, lo que te obligará a huir en muchas ocasiones cual Pacman con los fantasmas. Así que como vemos no es tan fácil; contamos con 3 vidas que perderemos con bastante facilidad sino sabemos jugar con inteligencia. En total hay 24 pantallas, y cada 4 entraremos en una habitación de bonus donde podremos obtener un objeto.

Majikazo es otra pequeña joya del software postcomercial y uno de los tres mejores juegos de Kralizec. Parece que la temática de la magia está muy presente en Kralizec, recordemos Dahku, por ejemplo. Y es que nos estamos acostumbrando a tener unos juegos impresionantes en los últimos tiempos. Así que ya sabemos lo que nos toca, mimar a los creadores felicitándoles para que sigan teniendo ánimos para futuras creaciones. Y un consejo, ¡Jugad, y jugad en un MSX real! Que en definitiva para eso se programan, no para coleccionarlos en la carpeta de roms del BlueMSX.

Majikazo se lleva una más que notable nota desde que puntuamos en esta revista en lo que a juegos postcomerciales se refiere, aunque *Bombjack* nos sorprendió más en su momento. Un juegazo para MSX1 que bien podría superar en muchos aspectos a programas de Konami entre los años 1984 y 1986. Tenemos que valorar más lo que tenemos en casa, el producto nacional, aquí hay gente con mucho talento. Ojalá algún día nos sorprendan con algún título más pretencioso para ver hasta dónde son capaces de llegar.

Pantalla de bonus

Jugabilidad :	8
Música :	9
SFX :	8
GFX :	8
Adicción :	8
Extras :	7
Total :	**8**

Ya disponible



Konami siempre ha despertado el interés en todos los usuarios. Muchos de los que han llegado a ser programadores han tratado de hacerle un homenaje, hemos visto remakes como Knightmare Gold, clónicos como Sir Dan y ahora Manbow 2 llega más lejos afirmando ser una secuela en toda regla. ¿Lo habrá conseguido? Vamos a echarle un vistazo a tan esperado juego.

Después de nueve interminables meses Manbow 2 fue presentado en la reunión holandesa de Bussum de octubre. La razón principal residía en el formato para la distribución del programa, el cual se decidió finalmente optar por la versión cartucho con chip SCC incorporado. Entonces podemos hacernos una idea de lo que ello conlleva, creación de carcasas, grabación del juego en sus respectivas placas y luego la caja de presentación, la cual el primer día de venta aún no disponía de su manual de instrucciones porque no dio tiempo a tenerlo preparado.

El caso es que al final el juego está ya a la venta desde hace unos meses y ya tenemos solucionado el tema del manual y de las etiquetas del cartucho, que en su día hizo especular a más de un usuario. Pero el tiempo pone las cosas en su sitio y ya tenemos el producto completo al 100%.



Pantalla digitalizada de presentación

Lo primero que nos llama la atención es el estuche. Una caja al más puro estilo Dinamic edición de lujo. La única pega es que el diseño no nos parece muy acertado. La ilustración peca de sosa y poco lograda, ya que los planetas que están por detrás de los soles se vienen hacia delante por poco dominio del color, aparte de no haber sabido escoger su ubicación. La parte trasera nos parece poco profesional, deberían haber añadido textos, darle más importancia a la publicidad del juego y no colocar capturas sin más. Tampoco estamos pidiendo un *package* de la calidad de los *Shockware* de Matra pero un mínimo de intención en el diseño sí, ya que en esta época todo el mundo tiene una vaga idea en llevar un photoshop para hacer un simple montaje con gusto.



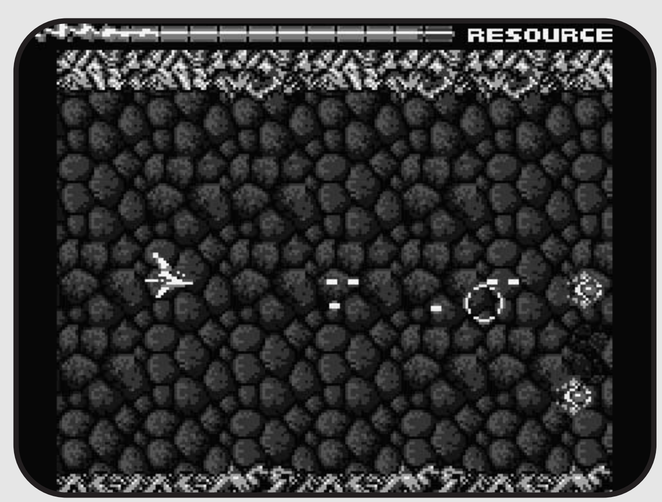
Fragmento de la intro

En fin, dejando a un margen el estuche nos metemos de lleno en el cartucho en sí. El cartucho es de nueva creación, con los moldes sacados de fábrica, el resultado impecable, precioso. El diseño de la pegatina está más conseguido que la caja. Decimos esto por la queja de algunos usuarios que creían que el cartucho era usado. Escuchadlo bien: ¡Los cartuchos son todos NUEVOS!.

Hacer una segunda parte de uno de los mejores juegos no sólo de Konami para MSX2, sino en general de la historia de los *shooters* en cualquier plataforma, se antojaba un tanto complicado. Al arrancar el juego nos aparece el bello logo del Manbow 2 en screen 8 después de un breve espacio de tiempo, algo inusual en un cartucho. Dejamos pasar un rato y comienza la intro la cual es interesante aunque preferiríamos ventanas más pequeñas con más animación a grandes franjas de pantalla con elementos diminutos. Percibimos un peor gusto en el tratamiento cromático que en el Manbow original y se nota lo que son gráficos imitando un referente y lo que es de cosecha propia, donde baja el listón un poco. Incluso la tipografía nos parece la misma del original. Nos preguntamos si Konami en una segunda parte hubiera mantenido el mismo estilo o lo hubiera cambiado. Recordemos que no tiene nada

que ver la tipografía del Gradius 3 con las anteriores entregas. En todo caso la intención es buena y se hace un esfuerzo por pretender ser una secuela lo más fiel posible. También remarcar el acierto en los fundidos de pantalla.

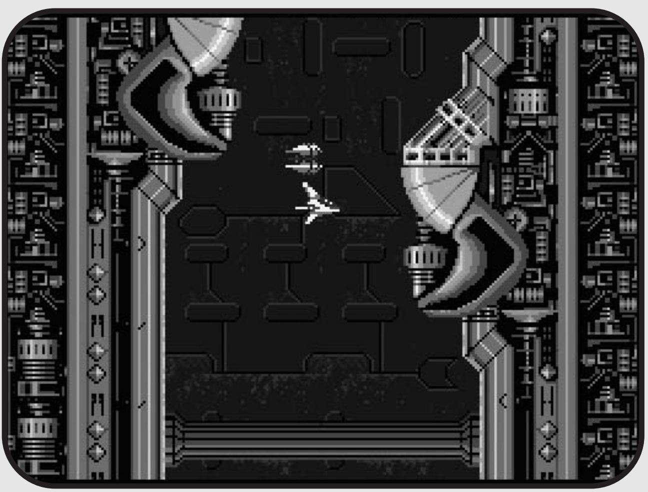
Hablaremos de los gráficos en general, no vamos a comentar pantalla por pantalla. Comenzaremos por el scroll, tenemos fases con scroll horizontal y vertical, como ya se hizo popular en la primera entrega. También combina ambos scrolls en una misma pantalla como ya hiciera su antecesor. Es un punto a favor para los chicos de **RenovatiO**. Además también es curioso como cuando llegamos a un Final Boss la pantalla de juego se expande a ambos lados con bastante picardía. Más detalles técnicos de scroll los vemos nada más empezar a jugar, si nos fijamos en el espacio exterior hay un bello efecto de parallax en las distintas velocidades de las estrellas. En este sentido se ha hecho un excelente trabajo, porque recordemos que al ser programado para MSX2 el scroll horizontal sigue siendo por software.



Primera fase

En primer lugar el juego está creado completamente en screen 5, al contrario que la primera entrega cuya pantalla de juego estaba realizada en screen 4. Aun así, Space Manbow que estaba más limitado cromáticamente hacía mejor uso del color, vamos, era magistral. En Manbow 2 se ha hecho lo que se ha podido dentro de sus posibilidades, pero una revisión así merecía un esfuerzo extra; es una lástima no estar a la altura del gran clásico. Es una cuestión de gusto, cuando ya está creada una pantalla y vemos todo en movimiento se percibe que al color le pasa algo, deberían haber vuelto a revisar los tonos RGB de los colores más problemáticos y sólo con haber subido un tono en algunos casos la cosa hubiera dado un vuelco espectacular.

Los decorados son detallados al igual que las naves y los ítems. Todo está bien trabajado, hasta las barras de energía. Pero eso sí, cuando el fondo es nítido el decorado del primer plano es confuso y viceversa, a veces tenemos unos primeros



Descendiendo en la segunda fase

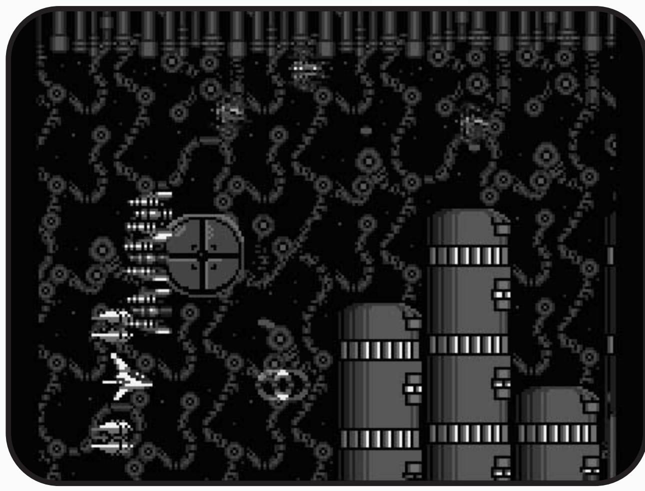
planos cojonudos, muy trabajados, con definición increíble y luego el segundo plano se muestra sucio, tan típico de los juegos holandeses. Tampoco parece acertado el abuso en colores chillones como el amarillo. En definitiva, es mejor el dibujo que el color.

Entramos ya a valorar la jugabilidad. Cabía esperar que por el tipo de juego y al ser creado en SC5 la suavidad iba a sufrir en el momento en que tuviéramos muchos enemigos en escena. Recordando otros juegos del género habidos para MSX2 que han hecho uso de Screen 5 (e incluso de bitmaps para algunos enemigos) nos viene a la mente rápidamente el Famicle Parodic 2 de Bit2 y su “lentitud”. Por suerte Konami siempre utilizó el SC2 en este tipo de juegos para culminar con el SC4 en el Manbow original, sabedores de problemas derivados del uso del modo con bitmap. Así que para empezar, el programador se ha complicado un poco la vida. Aun así ha salido bastante bien parado, ya que utiliza recursos para mantener la velocidad y el scroll a costa de perder animaciones en otros elementos o por reducir un pelín el tamaño de la pantalla. Por lo que podemos decir que estamos ante un programa bastante jugable y que controlamos bien la nave, con una suavidad agradable. Los enemigos tienen una velocidad envidiable, al igual que las estrategias de ataque, con movimientos a veces impredecibles; en este aspecto nos han sorprendido pero nos hubiera gustado ver más variedad de ellos o que si se muestran en la parte superior de la pantalla y nosotros estamos en la inferior, todavía puedan ponernos en un aprieto.

Y hablando de enemigos, ¿cómo es nuestro armamento? Ai igual que en Space Manbow el arma mejorará con la barra de power; deberemos coleccionar cápsulas rojas para ello. El nuevo armamento lo iremos consiguiendo al cambiar de fase al igual que la elección de nuevos continues o vidas.

La dificultad es ajustada, ni muy fácil ni muy difícil, depende claro está del arma que llevemos en cada situación.

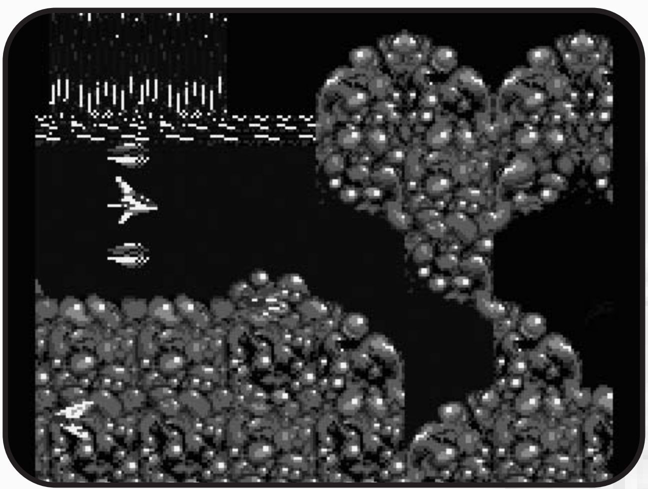
No es un juego que cuando has perdido todas las vidas apetece volver a empezar una y otra vez como sucedía en el original. Las malas lenguas dicen que el enemigo final es insufrible, casi imposible de derrotar.



Fase quinta

El tema sonoro en teoría debería ser el plato fuerte ya que se ha hecho un esfuerzo porque el cartucho llevara incorporado de serie el chip SCC. Puede parecer una pijada pero hay usuarios que no tienen cartuchos con chip SCC o a lo sumo uno. Para ellos es una bendición ya que en muchos casos se salvarán de pagar cantidades desorbitadas en subastas por un juego antiguo. La cuestión es si la calidad de los temas son suficiente como para haber invertido tanto esfuerzo en la incorporación del chip. Digamos que en líneas generales ha valido la pena, ya sabemos que nos va a sonar a Konami porque el sonido SCC tiene unos instrumentos muy típicos que lo han hecho "catalogable". Pero que te suene a Konami no quiere decir que las melodías estén a su altura. La verdad es que en líneas generales no lo están, Konami puso el listón muy alto y sus melodías eran impresionantes, nos podemos acordar ahora perfectamente de muchas de ellas sin tener el juego delante, han quedado en el recuerdo colectivo. En Manbow 2 son correctas salvo en alguna excepción en la que sí nos ha gustado mucho algún tema concreto. Para los poseedores de un Turbo R hay un problema importante de ajustes entre los volúmenes del SCC y el PSG, ya que mientras jugamos apenas percibimos los efectos de sonido porque quedan tapados por la melodía. Pero por fortuna tiene solución, si arrancamos el juego pulsando la tecla de función F1 nos aparecerá un menú que primero nos pide si queremos borrar las partidas grabadas y luego podemos ajustar los volúmenes por separado del SCC y el PSG, así que ya sabemos lo que toca, subimos el PSG y bajamos el SCC, así el sonido será perfecto y podremos disfrutar de los efectos de sonido. Estos efectos están a la altura de las circunstancias y celebramos la incorporación de voces digitalizadas como "Start" o "Continue" que dan un aire fresco al título.

Otro punto novedoso lo tenemos a la hora de continuar la partida. El cartucho es en sí una Megaflash Rom con lo cual graba partida en su propia Flash donde se graban nuestros progresos. Así una vez nos hayan eliminado, en la pantalla de presentación aparecerá la opción *Continue* la cual tiene hasta 3 partidas para salvar. Continuaremos a principio de la fase donde hayamos acabado con la posibilidad de elegir el armamento que teníamos en la partida anterior.



Pantalla final

En definitiva, un juego muy atrevido, sobretodo para ser la primera incursión del programador Norakomi en los videojuegos de MSX. Quizá le ha venido un poco grande el proyecto y le ha faltado pulir unos detalles que hubieran hecho de esta secuela un juego épico en la programación post comercial.

Aun así creemos que estamos ante uno de los cinco mejores juegos holandeses de todos los tiempos. Se ha echado valor y eso es digno de elogiar. Esperamos que tenga una buena acogida y ello anime a **RenovatiO** para seguir creando soft para nuestro sistema.

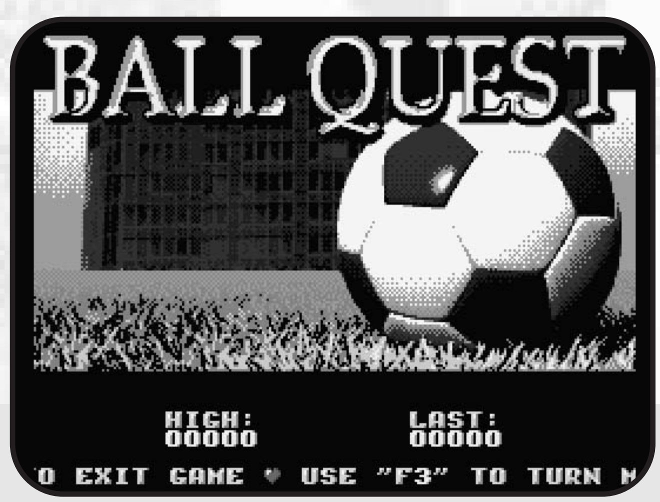
Jugabilidad :	8
Música :	7
SFX :	8
GFX :	7
Adicción :	7
Extras :	8
Total :	**8**



BALL QUEST

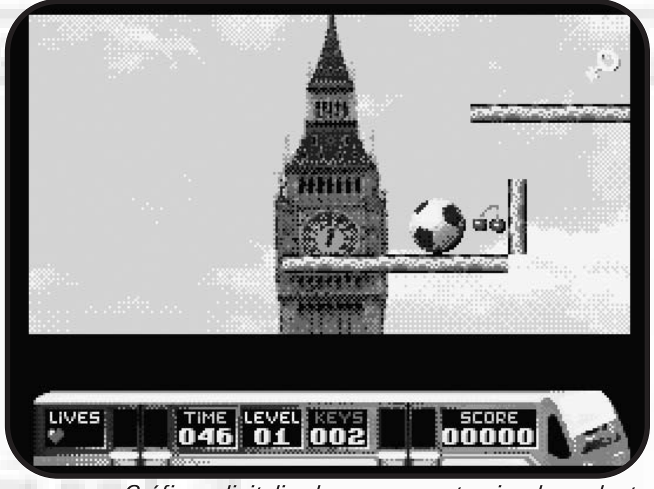
Este juego es toda una sorpresa, pues aparece en un concurso propuesto por el MSX Resource Center (Mega#Reto 1) y los creadores son programadores (**Shiru** y **Alone Coder**) habituales del sistema ZX Spectrum. Así que se han atrevido a versionar un juego creado para la máquina de Sinclair a MSX2 con un resultado más que notable conociendo de antemano cómo son estas “per-versiones”.

Estamos ante un plataformas atípico, más bien es un juego de habilidad. Debemos conducir un balón de fútbol hasta la salida; sí, es cierto, un poco surrealista el juego. Hay que acostumbrarse a los saltos del esférico para cogerle el truquillo. Si mantenemos pulsada la barra espaciadora el salto se realizará más alto. Hasta que nos hagamos con el control perderemos varias vidas y echaremos varias partidas.



Pantalla de título

El grafismo tiene buena presencia, superando en colorido a la versión Spectrum. Todo es digitalizado, desde un fondo immóvil hasta todos los sprites, pero tiene la virtud de que el primer plano, donde jugamos posee un scroll multidireccional y da una sensación de profundidad brutal. Para que nos hagamos una idea viene a ser algo parecido a la técnica del rendering usada en el juego Donkey Kong Country de Super Nintendo, es decir, gráficos digitalizados posiblemente creados en un programa de 3D. Claro, la pérdida de color en MSX respecto a una Super Nintendo es demoledora, pero han sabido crear una ambientación poco cargada para ser más



Gráficos digitalizados con apecto visual excelente

resultón. Las animaciones del balón (rotaciones) son geniales, al igual que otros elementos como el reloj de arena.

La música en PSG es más que correcta, tiene unos temas muy logrados y animados para los fans del sonido “electrónico” de los ordenadores 8 bit de los ochenta. Los efectos de sonido también están en la tónica de las melodías, nos recuerdan enormemente al sonido de la casa Gremlin (Auf Wiedersehen Monty).

Es un juego difícil, sobretodo por el control. Debemos mostrar gran habilidad para llegar a la salida sin antes caernos de las plataformas o ser aplastados por algún enemigo.

Un título muy recomendable, tremendamente técnico, aunque puede llegar a molestar la línea intermitente que aparece a un lado de la pantalla.

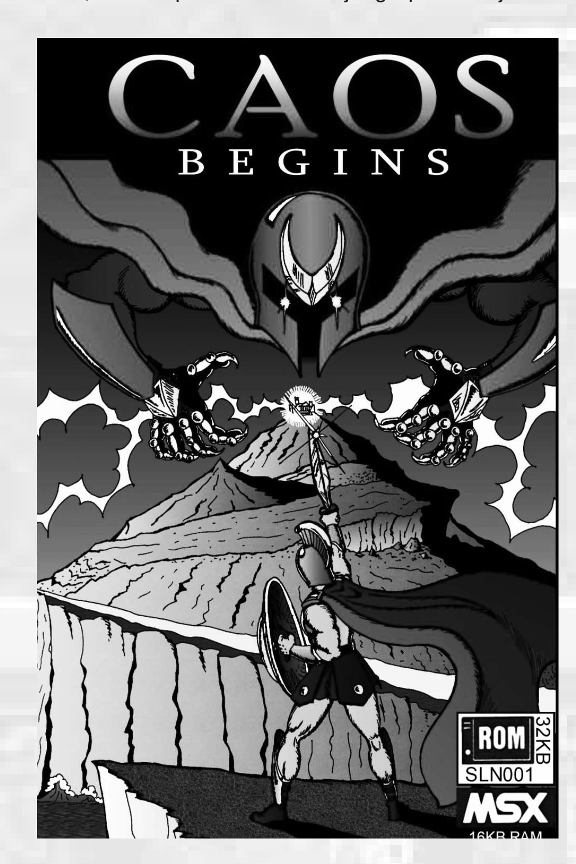
Jugabilidad :	7
Música :	7
SFX :	7
GFX :	7
Adicción :	7
Extras :	6
Total :	**7**

Ya disponible

CAOS BEGINS

Ha sido el acontecimiento de este principio de año en el mundillo MSX. Un juego que ha deslumbrado en el pasado concurso de MSXdev, llevándose el primer premio. Y como no podía ser menos en apenas dos meses nos sorprenden poniéndolo a la venta en formato cartucho, una gran iniciativa por parte de los creadores, ***Hikaru Games****.*

Lo primero que nos damos cuenta es que el formato no tiene nada que ver con los clásicos Shockware que puso de moda Matra y que han creado escuela. En este caso el formato nos recuerda enormemente a las cajas de cartón de los megarom Konami o de Hal Laboratory. Tiene la ventaja de que no se deteriora tanto con el uso, ya que éste era el gran inconveniente de los Shockware, que al ajustar tanto el tamaño del cartón con el del cartucho se forzaba el primero. Se agradece una ilustración dibujada de cosecha propia. Aunque tenga cierto aire "retro" se aparta de la moda de las portadas creadas por diseñadores gráficos. ¡Sí señor, cada uno a lo suyo! El diseño gráfico es más discreto, en este caso las producciones Shockware parecen estar más trabajadas. De todas maneras podemos decir que en general el package sale airoso, vale la pena tener este juego por la caja.



Una vez el cartucho insertado dejamos pasar la intro, la cual sólo es texto narrando la historia del juego (con fuentes propias) acompañado de un bello tema musical con una atmósfera muy oscura, mezclando sonidos del tipo intro Undeadline con otros más próximos al software europeo de las cintas de los años ochenta. La tipografía del logo está en consonancia con el estilo marcado por el Columns en su intento de emular el arte pelasgo o helénico, pero en cambio es radicalmente diferente al logo propuesto en la caja.

Pulsamos la barra espaciadora y comienza el juego con una breve y épica melodía introductoria. Sí señor, a primera vista tenemos la sensación de que estamos ante una adaptación del *Maze of Galious* en una pantalla más reducida. Movemos nuestro personaje y qué buena sensación, una suavidad espectacular y qué buenos espadazos. Pero dejemos la jugabilidad para más adelante.

El aspecto visual consigue un nivel considerable teniendo en cuenta que estamos ante un limitado MSX de primera generación. Fondos exquisitos, plataformas muy dignas (suelos, escaleras, columnas, animaciones de las antorchas, cascadas de agua, troncos de árbol...) y sprites monocromos de enemigos muy logrados con un aspecto profesional. La orla de la parte inferior donde se sitúan los iconos y la puntuación también tiene su gracia. En resumen, un juego muy trabajado gráficamente, con mucha fuerza, color y volumen, no tiene nada que envidiar a muchos juegos comerciales de la época dorada del MSX.



La música está a un nivel altísimo, cada mundo y cada momento tiene la suya propia. Carlo Bandini ha sabido crear unos temas heroicos que vienen que ni pintado. Aunque se perciben muchas influencias sobretodo de Konami y Compile, compañías maestras en el uso del PSG, Carlo sabe llevar a su

Caos Begins



terreno todos los temas con sello propio, al igual que hizo con Majikazo meses atrás. Parece mentira que con la limitación que ofrece el sencillo chip PSG donde todo "suena igual" porque no disponemos de instrumentos, podamos diferenciar a un músico de otro. Con Carlo esto pasa, lo cual quiere decir que estamos ante un músico personalísimo y que está en camino de hacer historia en nuestro sistema. El tema del castillo de Zineus es un buen ejemplo, épico para un juego de mitología, con una melodía virtuosa y con una base rítmica propia de heavy metal clásico ochentero. El tema de Hephaestus es otra pequeña joya que nos trae los mejores recuerdos de juegazos como Ys o Vampire Killer; vale la pena quedarse quieto un rato esperando en esa pantalla sin salir hasta poder apreciar ese maravilloso cambio que hace el tema en su parte final. En el jardín de Circe la melodía es muy fresca, recrea muy bien lo que es el aire libre y rompe con la tónica del interior del castillo.

La parte de programación corre a cargo de Toni Burguera que nos sorprende por ser su primer juego. Realmente lo ha bordado, cuesta creer que en su primera incursión como programador en ASM para MSX consiga crear un juego como éste. Como decíamos al principio *Caos Begins* sigue las pautas del *Maze of Galious* pero es más pequeño en todos los sentidos. El juego de Konami es una obra maestra y *Caos* es un muy buen juego. Aun así, *Caos* conserva una jugabilidad muy digna y es lo que hace que este juego sea especial. Un juego en que todos los elementos se mueven con suavidad, sin brusquedades y, además, lo mejor de todo es que crea adicción. Se acaba la partida y quieres volver a intentarlo. Ésa es la magia de un juego y *Caos* la tiene. Y si aparte los gráficos y la música son estupendos, ¿qué más se puede pedir?

El juego no es precisamente fácil, tampoco ayuda el que no hayan continues ni passwords. Todo reside en saber jugar, en no ir a lo loco, pues nos darán cuatro golpes y adiós muy buenas. Hay que usar con inteligencia el armamento que vamos consiguiendo y saber con qué enemigos nos enfrentamos. El conseguir experiencia es fundamental para poder afrontar nuevos enemigos más peligrosos. Y para no quedarnos encerrados y dar vueltas absurdas por el castillo debemos usar los ítems adecuados en cada sitio para que se nos vaya abriendo el camino (algunas tienen su gracia como las gafas de la verdad).

Concluyendo, es un pecado no disfrutar de este *Caos Begins*. Si os van este tipo de juegos no os lo deberíais perder. Lo único que se le podría achacar es que la versión cartucho fuera expandida, añadirle algún extra. Algunos echarán de menos una intro más elaborada con bellos gráficos, otros lamentarán no poder usar su FM... pero son adiciones que por su ausencia no desmerecen el resultado final. Tal como está sigue siendo un juegazo. Ahora depende de ti, si te quieres hacer con el cartucho creemos que vale la pena. Un juego así merece tener su cartucho y su package y presumir de él en nuestra estantería. Animamos a Hikaru Games a seguir en la onda y esperamos ansiosos una nueva producción. ¡Gracias por hacernos disfrutar!



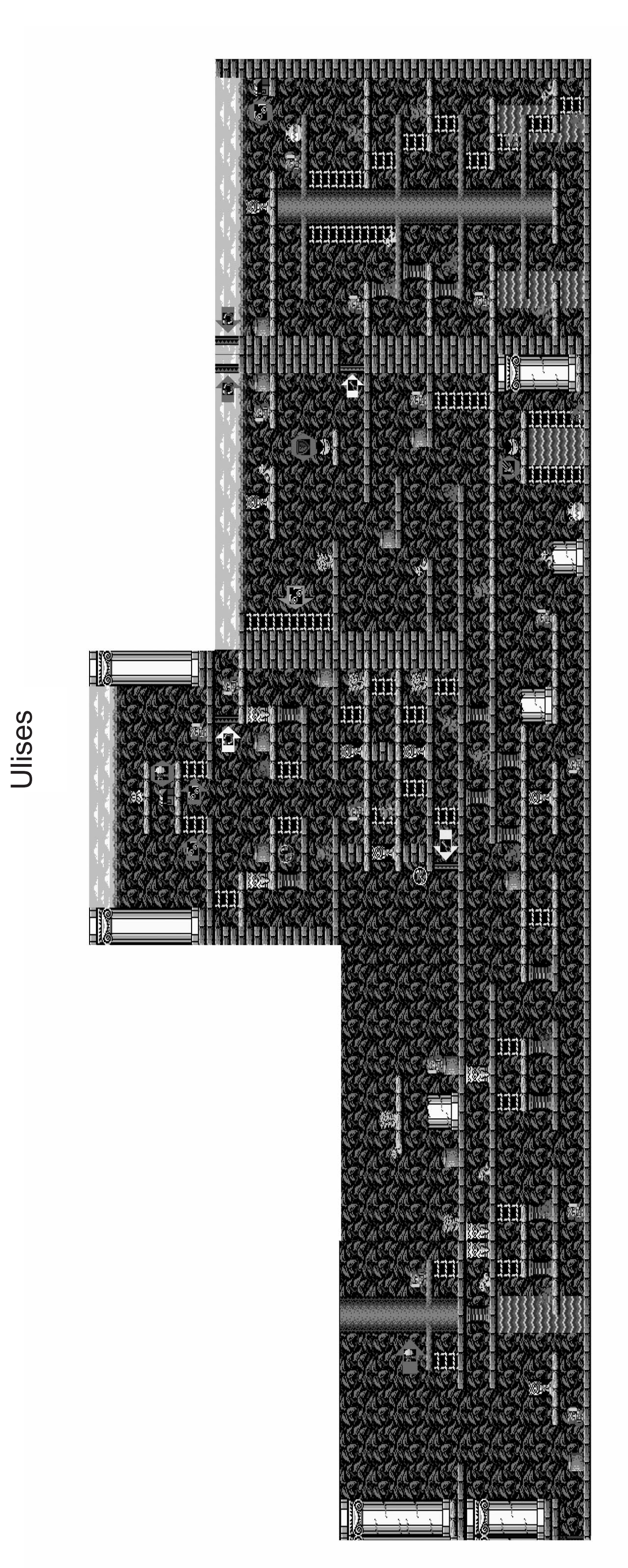
Jugabilidad :	8
Música :	9
SFX :	8
GFX :	8
Adicción :	9
Extras :	7
Total :	**8**

El castillo de Zineus

RUSTY KEY	ROLLING BOOTS
ROYAL SEAL	IVORY SHIELD
TRUTH GOGGLES	TITAN BRACELET
GOLDEN AXE	NEMESIS SHIELD
SILVER KEY	HECATE ARMOR
ROYAL SCEPTRE	MAGIC WAND
MAGIC SEED	ARES SWORD
PICCOLO	HERMES BOOTS
ORPHEUS LYRE	EBONY SHIELD
GOLDEN KEY	WOLF CLAW
PROMETEUS FIRE	SOLDIER SHIELD
PARCAE SCISSOR	GLADIUS

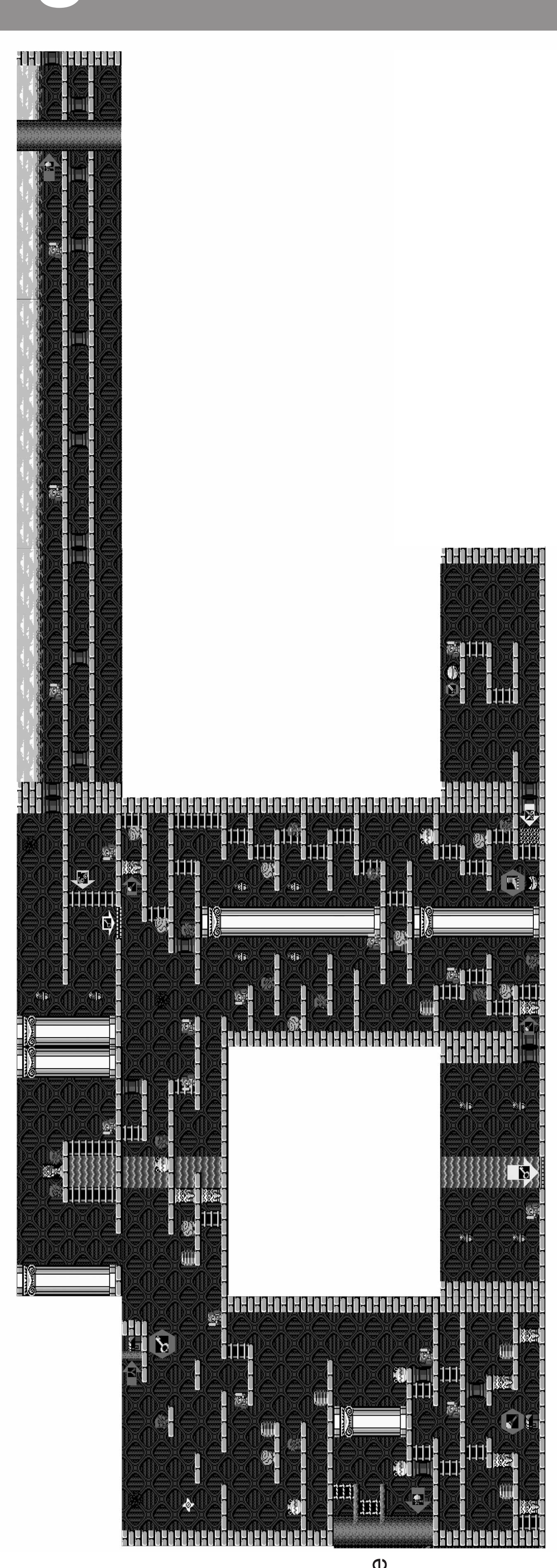


El jardín de Circe

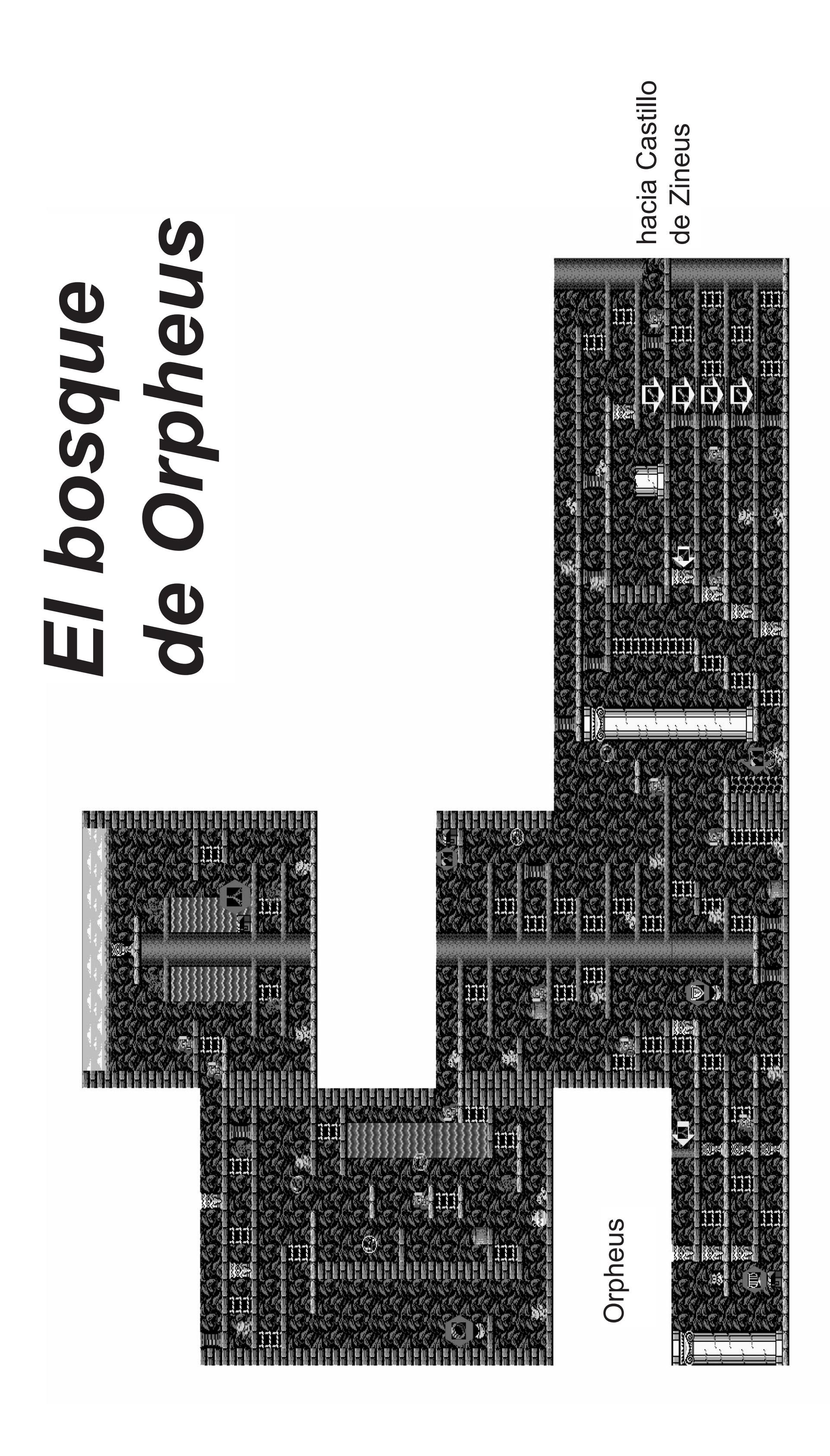


El castillo de Zineus

hacia Hephaestus

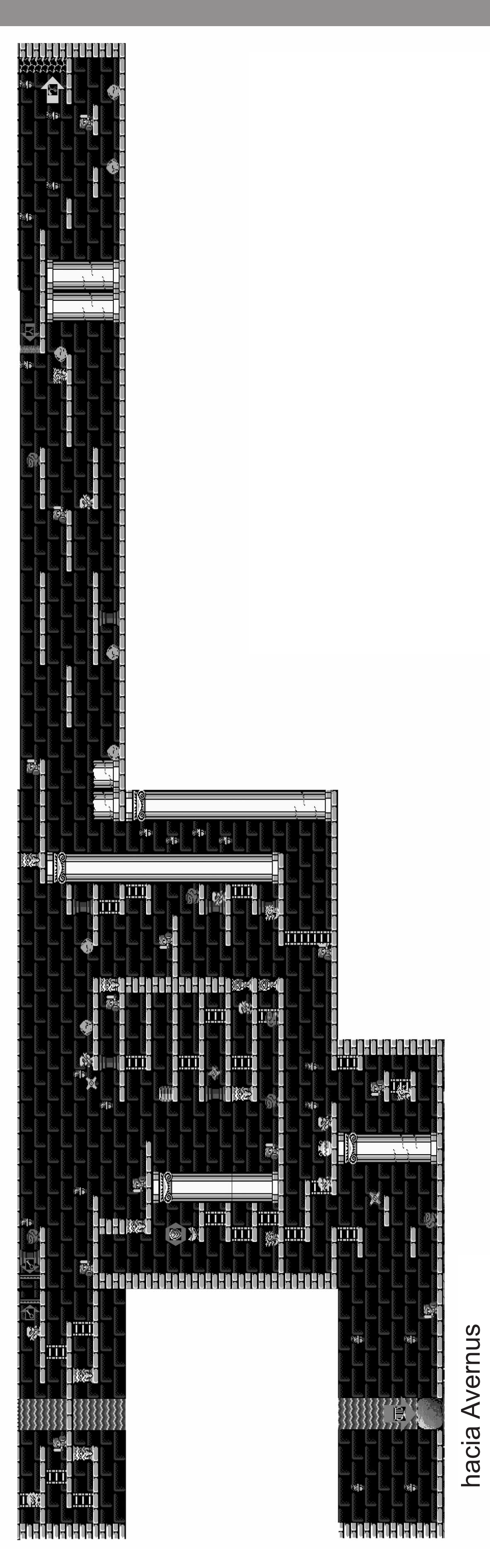


hacia Bosque de Orpheus

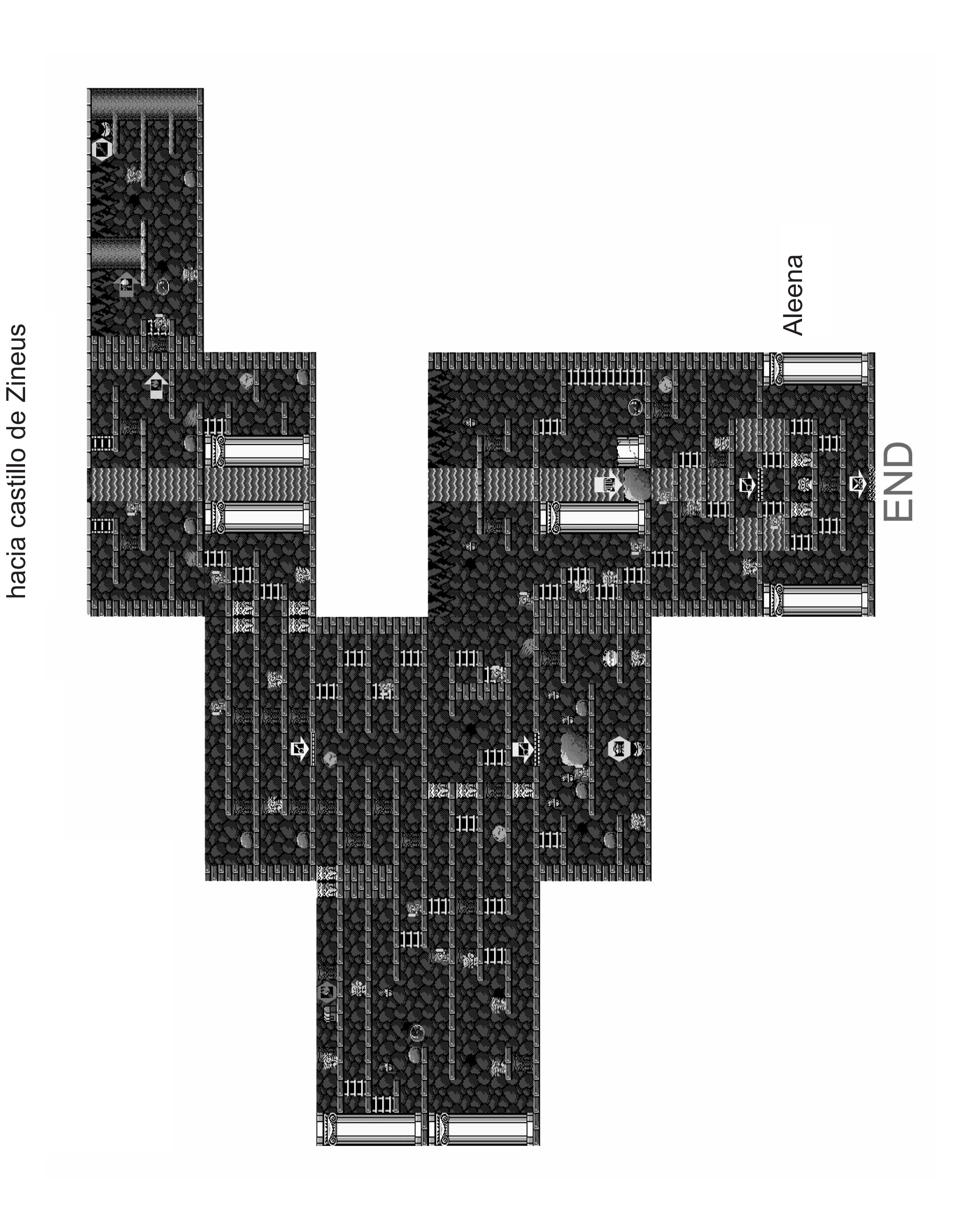

El bosque
de Orpheus
hacia Castillo
de Zineus
Orpheus

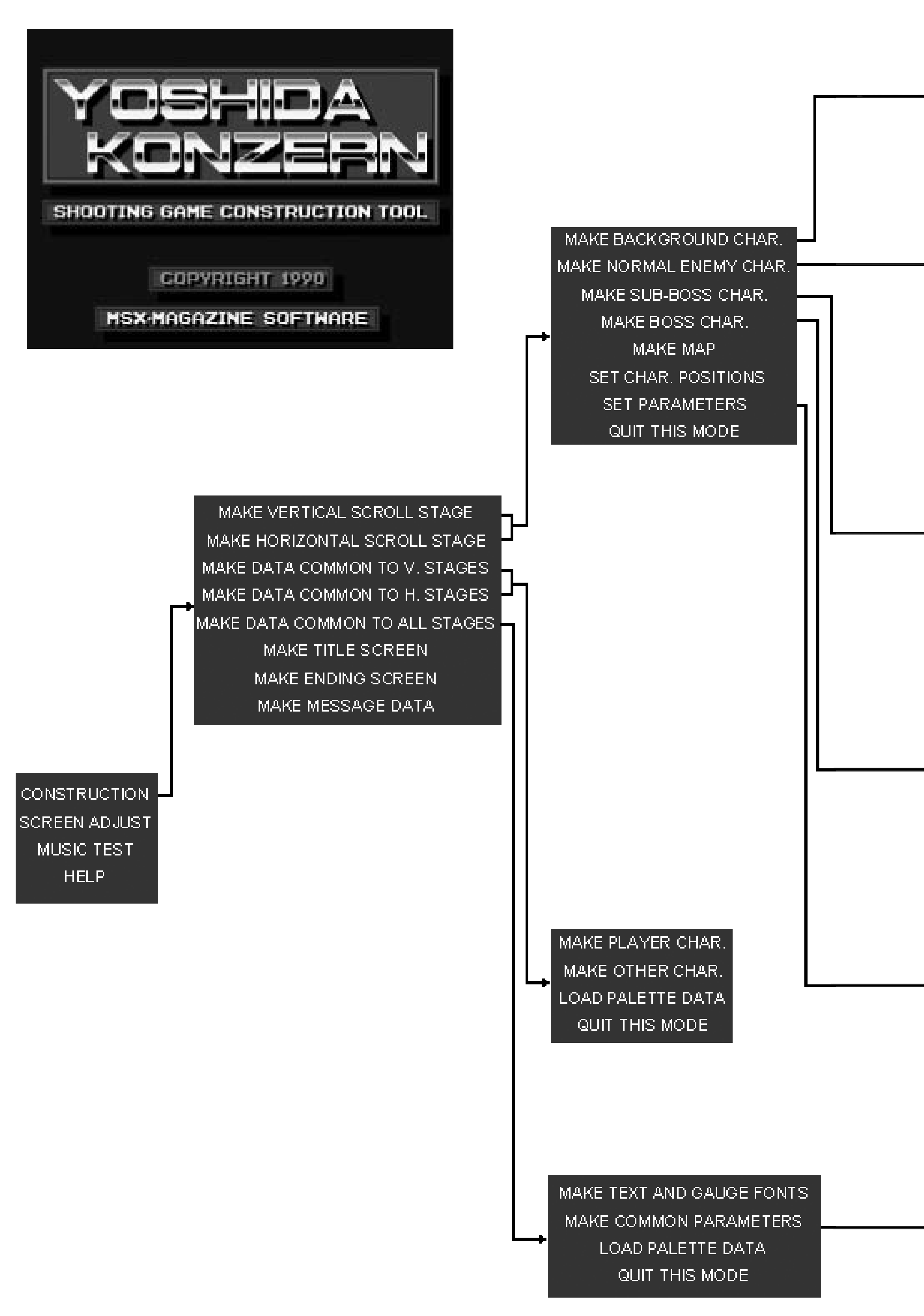
Mapa fotográfico

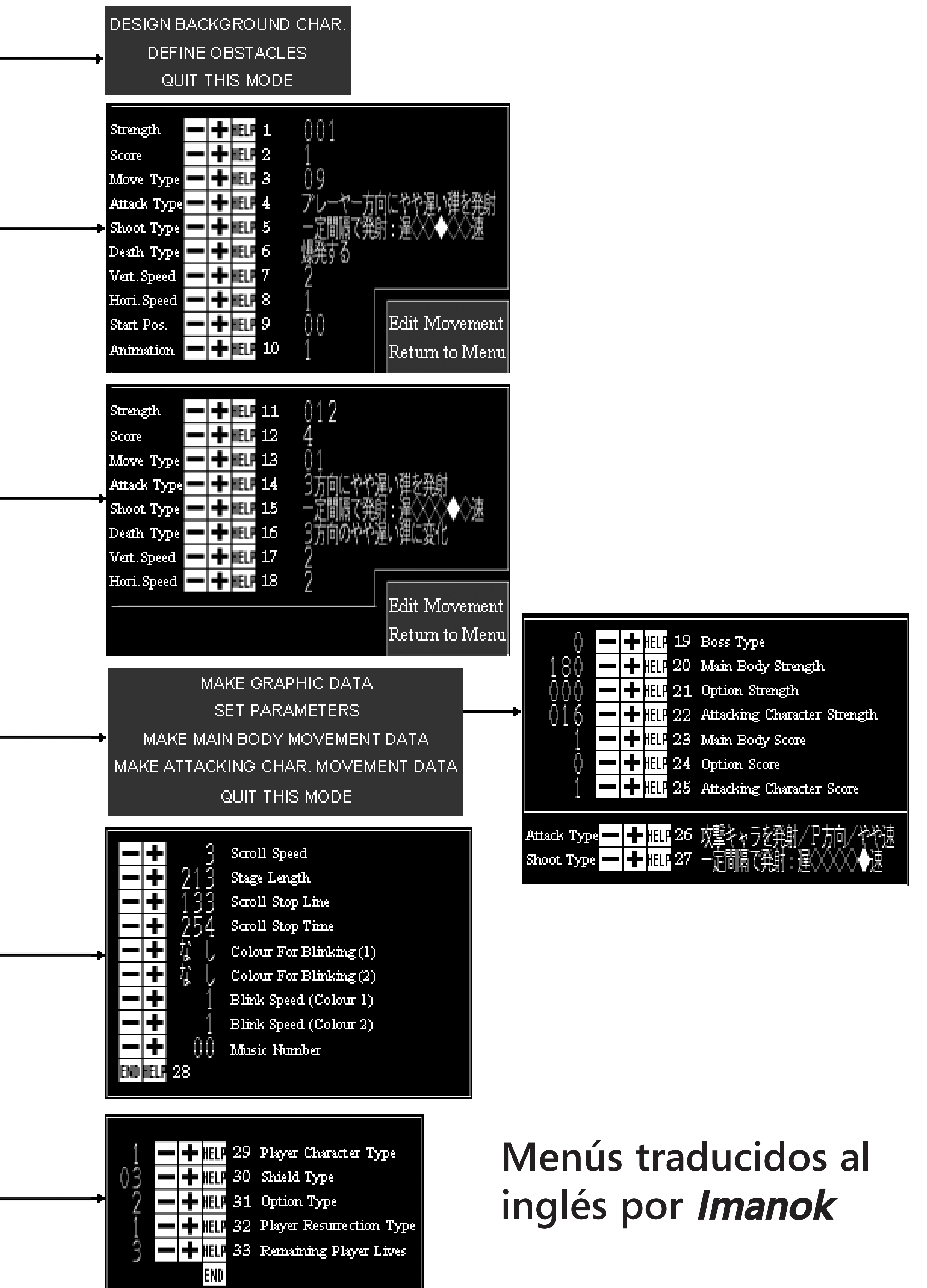
El castillo de Zineus



Avernus




YOSHIDA KONZERN
SHOOTING GAME CONSTRUCTION TOOL
COPYRIGHT 1990
MSX-MAGAZINE SOFTWARE
MAKE BACKGROUND CHAR.
MAKE NORMAL ENEMY CHAR.
MAKE SUB-BOSS CHAR.
MAKE BOSS CHAR.
MAKE MAP
SET CHAR. POSITIONS
SET PARAMETERS
QUIT THIS MODE
MAKE VERTICAL SCROLL STAGE
MAKE HORIZONTAL SCROLL STAGE
MAKE DATA COMMON TO V. STAGES
MAKE DATA COMMON TO H. STAGES
MAKE DATA COMMON TO ALL STAGES
MAKE TITLE SCREEN
MAKE ENDING SCREEN
MAKE MESSAGE DATA
CONSTRUCTION
SCREEN ADJUST
MUSIC TEST
HELP
MAKE PLAYER CHAR.
MAKE OTHER CHAR.
LOAD PALETTE DATA
QUIT THIS MODE
MAKE TEXT AND GAUGE FONTS
MAKE COMMON PARAMETERS
LOAD PALETTE DATA
QUIT THIS MODE



Menús traducidos al inglés por *Imanok*

La Memoria en el MSX

[HAGAMOS MEMORIA]

La memoria de nuestros MSX es, fundamentalmente, una de las partes más vitales e importantes del sistema. Lógicamente sola no sirve para nada, ya que sin un microprocesador que la maneje y utilice, no tiene ningún sentido. Pero así como hay partes del MSX que son perfectamente prescindibles para su funcionamiento, tales como un chip de sonido, una disquetera, etc... sin la memoria no tendríamos nada.

Al ser el MSX un ordenador personal incluso desde su versión más reducida (8k de Ram), cuenta con una cantidad de memoria bastante superior a cualquier videoconsola de la época, y que lo dota de la capacidad de hacer "algo más" que juegos. Le da la capacidad de tener un sistema operativo, un BASIC, utilidades, etc... algo impensable para otros sistemas de entretenimiento, incluso para otros ordenadores personales de la época.

Pero todo esto que suena tan bonito tiene un pequeño problema en MSX: La ubicación de la RAM es totalmente desconocida, en su gran mayoría, para el programador. Aunque es cierto que el sistema operativo de disco MSXDOS ya en su versión más vieja nos informa un poco de la ubicación de esta Ram, sólo con la aparición del MSXDOS2 tuvimos un gestor de RAM en MSX. E incluso éste, no es 100% perfecto pues sólo nos reconoce los denominados "memory mapper", inexistentes de fábrica en cualquier MSX de primera generación.

Realmente trabajar sobre MSXDOS2 tiene muchísimas ventajas para el programador ya que cuenta con un sistema operativo que realiza toda la faena sucia por él, como cualquier SO que se precie. Pero todo esto era inviable para la programación de utilidades/soft de entretenimiento ya que pocos podrían ser los privilegiados en tener DOS2 en sus máquinas, además de ser algo relativamente "moderno" en el estándar.

Por lo tanto, la ubicación de la memoria ha producido más de un dolor de cabeza a los programadores del estándar. Bueno, más bien a los programadores de juegos en casete. Ya que, salvo algunos *casos raros* ningún rom o megarom de MSX conocido ha usado jamás más de 8 ó 16k de Ram, y esa porción de memoria sí es conocida para el programador y además posicionada adecuadamente por el MSX al arrancar.

El MSX es y será un sistema versátil, muy versátil y permitía a cualquier fabricante posicionar la memoria de la forma que mejor le viniese. Es que, desde mi punto de vista, es una gran ventaja y aunque constituye sin duda lo más grande de nuestro sistema se ha convertido en un problema. Sobre todo en los últimos tiempos. Así que, intentaremos en este artículo ayudar a erradicar la gran mayoría de esos problemas. El artículo no pretende meterse en temas de memory mappers y sistemas operativos, hablaremos básicamente de búsqueda de memoria y obtención de slots/subslots de 64k básicas. Hablamos entonces de un artículo relacionado más con la primera generación.

[LEYENDAS URBANAS]

Como cualquier sistema que cuente con muchos años de existencia, el MSX cuenta con múltiples leyendas urbanas o falsos rumores (como queramos llamarlos) en muchos de sus ámbitos. Y la programación no podría estar exenta de ellos. No se puede acusar a nadie sobre este hecho. La información técnica del standard nunca ha sido algo al alcance de la mano hasta nuestra época actual. Incluso ahora mismo existen algunas lagunas para los no-japoneses. Pasemos pues a enumerar un par de esos conocidos casos.

* Si el slot de la memoria de la página 3 es xxxx, pongo ese slot en el resto de páginas del Z80 y ya tengo ubicadas las 64k:

FALSO. Aquí tenemos 3 problemas. Primero no sabemos si el equipo cuenta con 64k de Ram. Segundo que posicionando el slot en el resto de páginas, aun en el caso de que la memoria este toda en el mismo slot, podríamos obtener un bonito cuelgue ya que hay que tener en cuenta los SUBSLOTS. Y tercero, que, como se menciona antes, no sabemos si toda la memoria está en el mismo slot.

¿Porqué se ha extendido este rumor? Pues por 2 razones. La primera que en la gran mayoría de casos, efectivamente, la memoria está toda en el mismo slot. La segunda porque en MSX2 esto sí es funcional, excepto por el tema del subslot que "solucionábamos" con el famoso POKE -1, XXXX.

* La memoria mínima de un MSX es 16k de Ram.

FALSO. La memoría mínima de un MSX es 8k de Ram. Esa memoria se posiciona a partir de la dirección 0E000h. Y eso

es con lo único que podemos contar como realmente seguro.

De hecho, nos sorprenderíamos de la *gran cantidad de juegos* que existen para nuestro micro que solo hacen uso de esa cantidad.

[YA, PERO YO QUIERO USAR MÁS...]

Vaale. Que sí, que tú quieres usar más de 16k. No soy quién para impedírtelo, pero sí me veo obligado a ayudarte entonces en su búsqueda.

Antes de empezar, quisiera dejar claro que las rutinas que aquí se explican hacen uso de la rutina de búsqueda de memoria en un slot del driver de los interfaces de disco de MSX. Esto quiere decir que son bastante fiables. También he realizado pruebas en múltiples equipos con resultados satisfactorios. Pero no descarto, como siempre, algún problema puntual o algún error.

[DÓNDE ESTA LA MEMORIA, MATARILE RILE RILE]

Partamos de una base. Y como cualquier base para partir de ella, intentaremos que sea sólida. La BIOS ROM de nuestros MSX lo primero que hace, por necesidad propia, ya que tiene que posicionar sus "variables del sistema", es buscar la RAM para la página 3 del Z80, concretamente le interesa una pequeña zona que va desde 0F380h hasta 0FFFFh. Sí, en efecto, esto quiere decir que el sistema ya nos "roba" una importante cantidad de memoria, 3200 bytes. Y eso sin contar la posibilidad de tener disco y el más que necesario espacio de pila.

Ejemplo de carga de juego que requiere 64k

Entonces, sumando que la BIOS busca la memoria de la página 3 y que el ordenador con menos memoria disponible tiene 8k, pues ya sabemos como antes hemos dicho, que tenemos fijos para nosotros la memoria comprendida desde 0E000h hasta 0F380h, los nada despreciables 4992 bytes. Eso sí, hablando siempre de un programa ejecutado desde una ROM, y que no tenga necesidad de usar disco.

Que sí, que sí, que hablamos de usar más de 16k. Pero veamos qué tenemos. Tenemos una zona de memoria segura y un slot/subslot donde ya sabemos que hay RAM. Suficiente. Ya podemos empezar a buscar el resto de la memoria.

Lo primero que tenemos que ver es si el MSX cuenta con 16k. Si no tiene 16k, casi que buscar el resto es inútil. No he visto ningún documento que lo prohíba, pero no tiene mucho sentido dotar al MSX de 16k en un slot para una página no 3 y 8k en otro slot para la página 3. Ningún sentido. De hecho, lo QUE SÍ PUEDE PASAR es que el MSX tenga 8k de fábrica y se le ponga una expansión de memoria.

En ese caso, aunque tengamos que asegurarnos primeramente que el MSX tiene 16k, la BIOS del MSX, que es muy lista ya ha posicionado, seguro, en la página 3, el slot subslot con más memoria. Es decir, que si tiene uno de 16k y otro de 8k, tendremos ya posicionado el de 16k.

Volviendo a los casos lógicos de fabricación de MSX, si el MSX tiene 16k de Ram, los 8k extras están justo debajo de los 8k obligados. Es decir, que forman un solo bloque de 16k que empieza en 0C000h, hasta 0FFFFh, y haciendo la misma cuenta que antes con la memoria usada por la BIOS pues nos da de 0C000h hasta 0F380h. Impresionante. 13184 bytes para nosotros.

Sigamos con los casos normales y habituales en el estándar. 32k de Ram. Un ordenador con 32k de Ram tiene la RAM en dos bloques de 16k, no obligatoriamente en el mismo slot/subslot. Y esa memoria se posiciona desde 08000h hasta 0FFFFh.

Pero aquí ya empiezan los problemas. Necesitamos averiguar si existe o no y encima POSICIONARLA en la zona visible del Z80 para poder utilizarla. Ja. Ahí es nada. Eso sí,

de nuevo, hablando siempre desde un ROM. Porque, chica lista la BIOS y el intérprete BASIC ya nos la colocan. Cuando arrancamos el MSX y él mismo llega al BASIC y *siempre que el ordenador tenga 32k* ya tendremos posicionada esa memoria en el espacio visible del Z80. Y además si la cantidad de memoria que nos da la instrucción FRE(0), supera los 20k, ya sabemos que fijo que ese ordenador tiene 32k. Todo un avance, ¿no?

Más casos lógicos añadiendo memoria. 64k. Sip, paso a este modo porque *no tengo conocimiento de ningún MSX con 48k*. Eso no quiere decir que no existan, aunque pondría la mano en el fuego a que no existen desde ya. Pero en el hipotético caso de existir, esa memoria la tendríamos desde 04000h a 0FFFFh. Así es como se implanta la memoria en MSX, de "arriba a abajo". Pero vamos, ni caso, este MSX no existe. Será un Spectrum, fijo. A lo que íbamos, 64k. De nuevo, posibilidad de que esas 32k extra sobre un ordenador de 32k pueden partirse en dos bloques de 16k y estar cada bloque en un slot/subslot. Por si eso no fuese poco, el BASIC ya no las "ve". Y para rematar el único soft, clásico, que las usa y ve es el MSXDOS. Esto se anima.

Pero aquí no termina la cosa. Que en este pequeño repaso de implementación de memoria en un MSX *siempre* he dado por supuesto que cada extra se ha ido añadiendo en una página del Z80 diferente. ¿Qué quiere decir esto? Pues que el MSX puede tener, perfectamente 64k. Pero no 64k que vayan desde 0 a 0FFFFh, si no que puede tener 32k repartidas en dos slots/subslots diferentes y ambas en la zona de 08000h a 0FFFFh. ¡Toma lio! ¿Tiene el MSX 64K? Pues sí, si lo tiene, pero es que resulta que las tiene distribuidas de una manera que, actualmente y hasta donde yo conozco, NINGÚN soft va a utilizar correctamente.

[¿PERO NO BUSCAMOS 64K?]

Sí. Pero 64k no de cualquier manera. 64k paginados de 0 a 0FFFFH es lo que queremos. Y, encima, nos vamos a poner un poco pijos y también queremos saber esas 64k, hablando de direcciones son lineales, pero también si están en el mismo slot/subslot todas.

¿Qué queeé? Sí, ya he dicho que es pijotería, pero es algo MUY interesante para según que utilidades, saber si realmente la memoria está toda en el mismo slot/subslot. Os pongo el ejemplo clásico: un emulador de ROMS en RAM. Por fuerza necesita que toda sea memoria sea lineal, ya que emula una ROM que es líneal. Así que, si esa memoria no es lineal (hablando siempre en temas de slots/subslots), difícilmente funcionaría el ROM emulado en RAM. Es algo muy específico, lo sé, pero intentaremos que nuestras rutinas también nos informen de eso.

Aunque realmente para un soft nuevo que necesite más memoria de 8k/16k le es indiferente su ubicación.

[MENUDO ROLLO ME HAS SOLTADO Y TODAVIA NO ME HAS DICHO NADA]

Totalmente de acuerdo. Mucha explicación, mucho lío, seguramente estéis más liados que cuando empezásteis el artículo y encima no he dado ninguna solución. Me interesa. Si aprendéis todo, mis horas como colaborador de la revista están contadas. Fuera bromas, vamos ya con la parte programativa y seria. Explicación de la rutina que dios sabe dónde maquetarán los hacedores de esta revista, pero seguro que está cerca de este texto.

```
; *** MEMORY SUBROUTINES ***

RAM8K:          equ     0
RAM16K:         equ     1
RAM32K:         equ     2
RAM48K:         equ     3
RAM64K:         equ     4

; Especiales para lineal y carga de Roms

NORAML:         equ     0       ; No hay Ram para cargar algo de 16k
RAML16K:        equ     1       ; Podemos cargar algo de 16k
RAML32K:        equ     2       ; Podemos cargar algo de 32k linealmente
RAML48K:        equ     3       ; Podemos cargar algo de 48k linealmente

EXPTBL:         equ     0FCC1h
BOTTOM:         equ     0FC48h

RDSLT           equ     000Ch
WRSLT           equ     0014h
```

```
                    org         04000h
                    db          041h,042h
                    dw          initmain
                    ds          12
initmain:
                    di
                    im          1
                    ld          sp,0F380h
                    call        searchramnormal     ; Se busca Ram normalmente
                    call        searchramall        ; Se buscan TODA la Ram
                    call        searchramlineal     ; Se analiza si va linealmente

                    jp          $
```

En esta ocasión la rutina es muy sosa. No nos muestra nada en pantalla, ni suena, ni tenemos nada moviéndose en ella. Sencillamente rellena unas variables para que el programador sepa a que atenerse. Cada cual que aproveche estas rutinas como mejor pueda y las extienda. Después de la típica y obligada según las normas del estandard inicialización de la ROM, es decir, desconectar interrupciones, posicionar el modo de interrupciones 1 del Z80 e *inicializar la pila*, el programa llama a 3 rutinas que pasamos a comentar:

<**searchramnormal**> ¿A que mola mucho mezclar en las rutinas la mitad de cosas en inglés y la mitad en castellano? Pues nada, toca apechugar porque no se inglés y adquirí esa mala costumbre. A lo que vamos. Esta rutina es el pilar base de todas. Lo que hace es buscar por todos los slots/subslots del sistema 16k para cada pagina del Z80. Intenta buscar 64k lineales, es decir, que ocupen el rango de direcciones de 0 a 0FFFFh.

<**searchramall**> Para futuras aplicaciones. Busca TODA la Ram que tiene el MSX. Toda toda. Todos los segmentos de 16k que puede encontrar para todas la páginas del Z80 a excepción de la página 3. Por si alguien quiere, en un futuro, hacer alguna aplicación que utilice toda la RAM posible.

<**searchramlineal**> De la que hemos hablado unos párrafos antes. La rutina que busca que además de que la RAM sea lineal en rango de direcciones se encuentre toda en el mismo slot/subslot.

```
; *** BUSQUEDA NORMAL DE 1 SLOT CON RAM PARA CADA PAGINA ***

; ---------------------------
; SEARCHRAMNORMAL
; Busca la 64k de Ram
; Independiente slot
; ---------------------------

searchramnormal:
             ld          a,RAM8K
             ld          (ramtypus),a
             ld          a,(EXPTBL)
             ld          (rampage0),a
             ld          (rampage1),a
             ld          (rampage2),a
             ld          (rampage3),a

             xor         a
             ld          (ramcheck0),a
             ld          (ramcheck1),a
             ld          (ramcheck2),a
             ld          (ramcheck3),a

             call        search_slotram      ; Cogemos la Ram de sistema, porque
                                             ; el sistema ya entiende que es la mejor
             ld          a,(slotram)
             ld          (rampage3),a
```

```
            ; Comprobar 8k o 16k

            ld          c,0C0h
            call        checkmemdirect
            jr          c,searchramnormalend
            ld          a,RAM16K
            ld          (ramtypus),a

searchramnormal00:
            ; Buscamos Ram en las otras paginas

            ld          c,00h
            call        checkmem
            jr          c,searchramnormal40
            ld          (rampage0),a
            ld          a,1
            ld          (ramcheck0),a

searchramnormal40:
            ld          c,40h
            call        checkmem
            jr          c,searchramnormal80
            ld          (rampage1),a
            ld          a,1
            ld          (ramcheck1),a

searchramnormal80:
            ld          c,80h
            call        checkmem
            jr          c,searchramnormalend
            ld          (rampage2),a
            ld          a,1
            ld          (ramcheck2),a

searchramnormalend:
            ; Examinar la cantidad y apuntarla

            ld          a,(ramtypus)
            cp          RAM8K
            ret         z

            ld          a,(ramcheck2)
            or          a
            ret         z

            ld          a,RAM32K
            ld          (ramtypus),a
            ld          a,(ramcheck1)
            or          a
            ret         z

            ld          a,RAM48K
            ld          (ramtypus),a
            ld          a,(ramcheck0)
            or          a
            ret         z

            ld          a,RAM64K
            ld          (ramtypus),a
            ret
```

[SEARCHRAMNORMAL]

La primera de todas y rutina más importante de las 3, ya que las demás se nutren de sus resultados, es ésta. Al principio del artículo hemos hablado de las rutinas de una diskRom. Y ahora vamos a hablar un poco de ella, aunque no sea tema de este artículo, pues nos da una valiosa información. La diskRom necesita saber algunas cosas para inicializarse, a saber:

* Si el ordenador cuenta con 16k de Ram al menos: En caso contrario no se inicializa el driver. Un ordenador con 8k no soporta ningún driver de disco.

* Si hay 64k lineales para ejecutar el MSXDOS: De lo contrario no deja arrancar el MSXDOS y salta directamente al diskBasic.

Así que, por fuerza, una DiskRom busca memoria. Pero no toda. No. Se aprovecha de lo que la BIOS ya ha hecho por ella. En principio sólo busca memoria en dos páginas del Z80: La 0 y la 1. Esto es así porque aprovecha que la BIOS ya tiene que buscar esas páginas para el BASIC y las utiliza. No hace la faena dos veces. La diskROM tiene 4 variables que utiliza para almacenar el slot/subslot de esas 4 páginas, llamadas RAMADx, donde x es 0,1,2,3. Tanto 3 como 2 las rellena del slot/subslot posicionado en la inicialización de la diskrom, algo parecido a lo que hacemos nosotros con la página 3, que después veremos. Las otras 2, es decir, RAMAD0 Y RAMAD1, en principio las inicializa con el slot/subslot de la BIOS (EXPTBL). Y seguidamente busca la memoria, exactamente como nosotros para esas dos páginas.

A la hora de inicializar el MSXDOS comprueba si el valor de RAMAD0 y RAMAD1 es igual a EXPTBL, y si es diferente, inicializa el MSXDOS. Si es igual, no se ha encontrado memoria.

Esto puede ir un poco en contra de lo dicho al principio del artículo sobre el MSX y su versatilidad: No, no es 100% versátil. El código de la diskROM ya nos informa e indica que sí existen unas pequeñas normas. Y una de ellas es que NO es posible poner memoria en el slot de la BIOS para la página 0 y 1 del Z80. Cae por su propio peso, ya que, o está la BIOS o está la memoria, y sin BIOS no hay MSX. Pero, atención: SÍ es posible poner memoria en el slot de la BIOS para la página 2 y 3. Y no es que sea posible, es que hay un modelo de MSX conocido que así lo tiene: Toshiba HX20, de 64k de RAM. La configuración de memoria de este ordenador es, sin lugar a dudas, de las más curiosas conocidas. Dispone de 16k para la página 0 en slot 3-x (expandido), otras 16k para página 1 en slot 3-x (expandido), así que YA cumple la norma, PERO la memoria para página 2 y 3 está en el slot 0, vamos el mismo de la BIOS.

Esta pequeña historia viene a confirmar lo que ya hemos dicho: Las posibilidades del MSX son muchas y variables. Aunque imagino que los dueños de este ordenador, sobre todo Europeos, no muestren ningún síntoma de alegría por esto. Estoy muy seguro de que la gran mayoría de juegos en cassette Europeos no funcionaban en su máquina.



Monty también requiere 64k

Y ahora, SÍ, con toda esta información podemos pasar a explicar las rutinas. Lo primero que hace la rutina es inicializar nuestras variables:

[**ramtypus**] Contiene al final de la rutina el tipo de RAM que ha encontrado en el sistema:

0 Ram8K
1 Ram16k
2 Ram32k
3 Ram48k
4 Ram64k

La ponemos a 0 para empezar, es lo único que sabemos seguro, que nuestro ordenador cuenta con 8k de RAM.

[**rampage0,1,2,3**] las inicializamos con el slot de la BIOS, para ser igual de chulos que una diskRom (aunque luego hagamos la comprobación de diferente forma). Estas variables contienen al final de la rutina el slot/subslot donde ha encontrado 16k de Ram para esa página del Z80. Siempre y cuando hagamos caso antes a ramtypus. Explicación: Si ramtypus nos dice que el ordenador es de 16K, el contenido de rampage0,1 y 2 nos es indiferente, pues no ha encontrado más RAM.

[**ramcheck0,1,2,3**] Lo utilizamos para comprobar si en alguna de esas páginas se ha encontrado RAM.

Seguidamente, la rutina llama a una subrutina llamada <**search_slotram**>. Esta rutina lo único que hace es averiguar el slot/subslot posicionado en la página 3. Si hacemos memoria (nunca mejor dicho), ya hemos comentado que de lo único que podemos estar seguros al ejecutarse esta ROM es de que en la página 3 ya tenemos RAM. Entonces, sin ningún tipo de búsqueda especial rellenamos la variable [rampage3] pues la BIOS ya ha hecho eso por nosotros.

```
; ---------------------
; SEARCH_SLOTRAM
; Busca el slot de la ram
; Y almacena
; ----------------------

search_slotram:
             call         0138h
             rlca
             rlca
             and          3
             ld           c,a
             ld           b,0
             ld           hl,0FCC1h
             add          hl,bc
             ld           a,(hl)
             and          080h
             or           c
             ld           c,a
             inc          hl
             inc          hl
             inc          hl
             inc          hl
             ld           a,(hl)
             rlca
             rlca
             rlca
             rlca
             and          0Ch
             or           c
             ld           (slotram),a
             ret
```

Pero continuamos nuestra rutina. Lo siguiente que tenemos que averiguar es si el ordenador es de 8 ó 16k. Ya que, en efecto, tenemos el slot/subslot de la página3, tenemos ya el primer slot con memoria pero... ¿tenemos más? No lo sabemos, así que hacemos ya la primera comprobación "hard" y es buscar memoria en la zona de 0C000h en adelante. Usamos para ello la rutina <*checkmemdirect*>.

<**checkmemdirect**> Clásica entre las clásicas. Y seguro que todo programador MSXero que lleve tiempo con la máquina alguna vez se ha encontrado con ella. La rutina hace lo siguiente: Lee un byte de una dirección y lo complementa. El byte es guardado en otro registro. Seguidamente escribe el byte complementado en esa misma dirección y vuelve a leer el valor y lo almacena en otro registro. Por motivos de seguridad, vuelve a complementar el valor (dejándolo como estaba al principio) y lo vuelve a posicionar en la posición de memoria. Y por último la comprobación. Si el valor del primer registro que hemos guardado (el complementado después de leido) es igual al valor del segundo registro guardado (el que no ha sido complementado pero ha sido leído después de escrito complementado (vaya galimatías)) es igual, significa que hemos podido modificar el valor de la posición de memoria. ¿Qué significa eso? Pues que estamos ante un byte que PUEDE ser de RAM. En caso contrario la rutina pone el Carry a 1 y regresa, es decir, indica error. Pero si esto no ocurre, decrementa el contador y repite la operación en la posición siguiente. En total comprueba 16 posiciones. ¿Suficiente? Pues sí, hasta el momento de escribir estas líneas ha sido suficiente, es decir, desde 1983 hasta hoy con esto ha sobrado para comprobar y buscar memoria.

```
; --------------------------
; CHECKMEMDIRECT
; Chequea si hay memoria
; En pagina C
; Y 16 posiciones por arriba
; ---------------------------

checkmemdirect:
             ld           h,c
             ld           l,010h
checkmemdirect0:
             ld           a,(hl)
             cpl
             ld           c,a
             ld           (hl),a
             ld           a,(hl)
             ld           b,a
             cpl
             ld           (hl),a
             ld           a,b
             cp           c
             jr           nz,checkmemdirectno
             dec          l
             jr           nz,checkmemdirect0
             or           a
             ret
checkmemdirectno:
             scf
             ret
```

Si esta comprobación no nos da error ya nos encontramos ante un equipo con al menos 16k de RAM. En caso contrario la rutina llama a su fin, y nuestra búsqueda se da por acabada.

"Pero si me dijiste al principio que podría ser que el ordenador tuviese 8k en un slot y una ampliación de memoria en otro, ¿por qué abandonas?" Pues porque, como también dije al principio, la BIOS *ya se ha encargado de posicionar en la página 3 el slot/subslot de la memoria más grande que ha encontrado*. Así que continuar es inútil. Pero la rutina continúa si tenemos 16k detectadas. Tanto <*searchramnor-*

mal00> como <*searchramnormal40*> y <*searchramnormal80*> realizan la misma operación, excepto que cada una busca memoria para página 0, 1 y 2 respectivamente. Aquí se hace uso de otra rutina: <*checkmem*>

<**checkmem**>: Esta rutina recorre todos los slots/subslots del MSX, haciendo uso de la megaarchiconocidayvenerada rutina SIGSLOT de mi maestro el gran KonamiMan. Como la rutina se sale del artículo no la vamos a comentar. Para cada slot/subslot dado por SIGSLOT llamamos a <*checkmemgen*>. Si <*checkmemgen*> devuelve el carry a 1, sigue buscando la rutina <*checkmem*> hasta que se finalizan todos los slots/subslots a buscar. Si encuentra memoria, es devuelta a la rutina que la llamó (<*searchramnormalxx*>) para que anote el slot/subslot donde encontró memoria.

```
; ---------------------
; CHECKMEM
; C : Page
; Cy : NotFound
; ----------------------

checkmem:
             ld          a,0FFh
             ld          (thisslt),a
checkmem0:
             push        bc
             call        sigslot
             pop         bc
             cp          0FFh
             jr          z,checkmemend
             push        bc
             call        checkmemgen
             pop         bc
             ld          a,(thisslt)
             ret         nc
             jr          checkmem0
checkmemend:
             scf
             ret
```

<**checkmemgen**>: La rutina es *exactamente igual* a la ya explicada <*checkmemdirect*>. Igual, con una sola diferencia. Hace uso de las rutinas de la bios RDSLT y WRSLT para leer y escribir el valor que antes escribíamos y leíamos directamente. ¿Por qué? Pues porque no hemos posicionado el slot/subslot para poder leerlo. Y además, ni falta que nos hace. Ya tenemos esas rutinas BIOS que lo hacen por nosotros, ¿no?

```
; -------------------------
; CHECKMEMGEN
; C : Page
; A : Slot FxxxSSPP
; 00 : 0
; 40:  1
; 80 : 2
; Returns :
; Cy = 1 Not found
; ------------------------------

checkmemgen:
             push        bc
             push        hl
             ld          h,c
             ld          l,010h
checkmemgen1:
             push        af
             call        RDSLT
             cpl
             ld          e,a
             pop         af
             push        de
             push        af
             call        WRSLT
             pop         af
             pop         de
             push        af
             push        de
             call        RDSLT
             pop         bc
             ld          b,a
             ld          a,c
             cpl
             ld          e,a
             pop         af
             push        af
             push        bc
             call        WRSLT
             pop         bc
             ld          a,c
             cp          b
             jr          nz,checkmemgen2
             pop         af
             dec         l
             jr          nz,checkmemgen1
             pop         hl
             pop         bc
             or          a
             ret
checkmemgen2:
             pop         af
             pop         hl
             pop         bc
             scf
             ret
```

Y por fin, para terminar ya la rutina principal <*searchramnormal*> se hacen una serie de comprobaciones. Si ramtypus = RAM8K, pues nada, adiós, no había más. Si ramcheck2 es diferente de 0, es que ha encontrado algo. Si no, pues fuera, es decir, tenemos 16k. Si hay algo, pues ya podemos asumir que tenemos 32k. Si ramcheck1 es igual a 0, no tenemos más, en caso contrario podemos afirmar que nuestro ordenador cuenta con al menos 48k de RAM. Algo, como ya hemos dicho raro o inexistente. Para finalizar, si ramcheck0 es igual a 0, pues acabamos de encontrar lo más raro del mundo, es decir, un ordenador de 48k. Pero como seguro que eso no es así, ya podemos decir que POR FIN tenemos nuestro ordenador con 64k. Misión cumplida.

```
; *** BUSQUEDA DE TODOS LOS SLOT CON RAM PARA CADA PAGINA ***

;---------------------
; SEARCHRAMALL
; Busca la Ram, y analiza
; Para buscar toda
; -----------------------

searchramall:
            ld          c,00h
            ld          hl,rambuffer0
            ld          de,ramcounter0
            call        searchramallgen

            ld          c,40h
            ld          hl,rambuffer1
            ld          de,ramcounter1
            call        searchramallgen

            ld          c,80h
            ld          hl,rambuffer2
            ld          de,ramcounter2
            call        searchramallgen
            ret

; HL : Buffer
; DE : Contador
; C: pagina

searchramallgen:
            ld          a,0FFh
            ld          (thisslt),a
            xor         a
            ld          (de),a

searchramallgen0:
            push        hl
            push        de

searchramallgen1:
            push        bc
            call        sigslot
            pop         bc
            cp          0FFh
            jr          z,searchramallgenend
            call        checkmemgen
            jr          c,searchramallgen1
            pop         de
            pop         hl
            ld          (hl),a
            inc         hl
            ld          a,(de)
            inc         a
            ld          (de),a
            jr          searchramallgen0

searchramallgenend:
            pop         de
            pop         hl
            ret
```

[SEARCHRAMALL]

Esta segunda rutina lo que hace es buscar todos los slots/subslots que contienen memoria para cada página, excepto para la página 3, cosa que explicaremos más adelante.

Las variables utilizadas en esta rutina son las siguientes:

[**rambuffer0,1,2,3**]. Dado que sumando slots y subslots, la cantidad de los mismos máxima en MSX son 16, estas variables apuntan a un buffer de 16 bytes. En ellos iremos almacenando todos los slots/subslots que contengan memoria.

[**ramcounter0,1,2,3**]. Un byte por variable. Guarda el número slots/subslots de memoria encontrados para cada página.

Poco más tiene esta rutina que no hayamos ya explicado. Hacemos uso de nuevo de *SIGSLOT* y vamos recorriendo todos, almacenando cada slot/subslot encontrado e incrementado *ramcounterx* en uno cada vez que detectamos memoria.

Esta rutina puede ser muy interesante para cualquier aplicación que necesite usar el máximo de memoria posible.

```
; *** ANALISIS PARA INTENTAR POSICIONAR EL MAXIMO DE RAM LINEALMENTE ***
```

[SEARCHRAMLINEAL]

La última rutina hace lo que también hemos comentado ya. Busca en todo lo encontrado por <*searchramall*> una posible configuración de máxima RAM lineal. Quizás la rutina más enrevesada de todas. Esta rutina es, lógicamente, más lenta pues tiene que recorrer todo lo encontrado y hacer muchas comprobaciones. No vamos a desmenuzar esta rutina y la dejamos como ejercicio para el lector. Seguro que existen otros métodos para conseguirlo mucho más rápidos, pero en este caso no es una rutina que requiera rapidez ya que se ejecutaría una sola vez al principio de nuestro programa.

```
; --------------------
; SEARCHRAMLINEAL
; Busca cuanta ram
; Tenemos en un slot
; linealmente
; Hace uso de:
;             SEARCHRAMNORMAL
;             SEARCHRAMALL
; --------------------

searchramlineal:
              ld          a,(rampage3)
              ld          (ramlineal3),a
              ld          a,NORAML
              ld          (ramlineal),a
              ld          a,(ramtypus)
              cp          RAM8K
              ret         z                       ; 8k solo
              cp          RAM16K
              ret         z                       ; 16k solo
              ld          a,(rampage2)
              ld          (ramlineal2),a
              ld          a,RAML16K
              ld          (ramlineal),a
              ld          a,(ramtypus)
              cp          RAM32K
              ret         z
              cp          RAM48K
              jp          z,searchramlineal48k
              ;  64k a investigar

searchramlineal64k:
              ; Comprobamos que los defectos
              ; son iguales

              ld          a,(rampage2)
              ld          (ramlineal2),a
              ld          (ramlineal1),a
              ld          (ramlineal0),a
              ld          hl,rampage1
              cp          (hl)
              jr          nz,searchramlineal64k0     ; no es
              ld          hl,rampage0
              cp          (hl)
              jr          z,searchramlineal64kend
              ; Miramos si los tres tienen 1 y se pira

              ld          a,(rambuffer2)
              cp          1
              jr          nz,searchramlineal64k0
              ld          hl,rambuffer1
              cp          (hl)
              jr          nz,searchramlineal64k0
              ld          hl,rambuffer0
              cp          (hl)
              jr          nz,searchramlineal64k0
              ret                                 ; Los tres tienen 1 y no era la
                                                  ; misma fuera
```

Los resultados son almacenados también en variables y nos indicarán lo siguiente:

[**ramlineal**] Puede tomar 3 valores:

0 NORAML, es decir no hay Ram suficiente para cargar algo de 16k.

1 RAML16K, podemos cargar algo de 16K. 2 RAML32K, podemos cargar algo de 32k.

3 RAML48K, podemos cargar algo de 48k.

[**ramlineal0,1,2**] nos informarán del slot/subslot donde se encuentra esa ramlineal encontrada.

```
searchramlineal64k0:
            ; Esto va a ser superespectacular y no va a ir ...

            ld          a,(ramcounter2)
            ld          b,a
            ld          hl,rambuffer2

searchramlineal64k1:
            push        bc
            push        hl
            ld          a,(ramcounter1)
            ld          b,a
            ld          a,(hl)
            ld          hl,rambuffer1
            call        searchramlinealgen
            pop         hl
            pop         bc
            jr          nc,searchramlineal64k2

searchramlineal64k11:
            inc         hl
            djnz        searchramlineal64k1
            jp          searchramlineal48k

searchramlineal64k2:
            ; Buscar en el 3

            push   hl
            push   bc
            ld          a,(ramcounter0)
            ld          b,a
            ld          a,(hl)
            ld          hl,rambuffer0
            call        searchramlinealgen
            pop         bc
            pop         hl
            jr          c,searchramlineal64k11

            ; Bingo!

searchramlineal64k3:
            ld          (ramlineal2),a
            ld          (ramlineal1),a
            ld          (ramlineal0),a

searchramlineal64kend:

            ld          a,RAML48K
            ld          (ramlineal),a
            ret

; Analisis de 48k
; Mirar si coinciden
; la pagina 2 y pagina 1

searchramlineal48k:
            ld          a,(rampage1)
            ld          (ramlineal1),a
            ld          hl,rampage2
            cp          (hl)
```

"¿Cómo? ¿Qué pasa con las cosas de 64k?" Vale, vale, tranquilidad. Repasemos otra vez la idea de la utilidad de esta rutina. Como comentamos al principio lo que aquí buscamos es la máxima cantidad de ram lineal (es decir, a ser posible de 0 a 0FFFFh) y que se encuentre en el mismo slot/subslot en todas las páginas.

Si volvemos a la utilidad de ejemplo, para cargar algo de 16k de Ram, pues necesitamos tener 32k de Ram, mínimo. ¿Por qué? Pues porque entendemos que la página 3 es *intocable*. Los juegos en ROM contienen código (a excepcion de ROMS de 64k muy raros) en la página 3. Así pues, un juego de 32k de ROM, se ubica, normalmente en la zona 04000h-0BFFFh. Y la página 3 (0C000h-0FFFFh) es donde se ubican las variables del juego. Entonces, necesitaremos tener 48k de Ram en nuestro MSX por esa regla de 3. La pagina 3,2,1. Así que, en caso de tener 48k (el caso imposible y raro) pues la rutina devolverá en [ramlineal] un 2, es decir, que se nos permite cargar cosas de 32k.

El mismo caso, para cosas de 48k. Si queremos meter una ROM de 48k, pues necesitamos 64k, por fuerza. ¿Y para cargar cosas de 64k? Pues si, es posible, nada es imposible. Pero para eso necesitamos algo demasiado enrevesado. 4 slots/subslots lineales (un por página) mas un slot/subslot para la página 3 donde contengamos la memoria. Vamos, que esto pasa obligatoriamente por tener una memoria externa. Y es demasiado complejo de montar. Por fortuna no se prodigan los juegos de 64 K ROM no mapeados.

```
            jp          z,searchramlineal48kend   ; Primero a ver si las 2 paginas
                                                  ; encontradas son iguales y evitamos
                                                  ; busquedas
            ld          a,(ramcounter1)
            ld          hl,ramcounter2
            cp          1
            jr          nz,searchramlineal48k0
            cp          (hl)
            ret         z                         ; Si solo hay 1 en ambos y no era la
                                                  ; misma, fuera, no hay

searchramlineal48k0:
            ld          a,(ramcounter1)
            ld          b,a
            ld          hl,rambuffer1

searchramlineal48k1:
            push        hl
            push        bc
            ld          a,(ramcounter2)
            ld          b,a
            ld          a,(hl)
            ld          hl,rambuffer2
            call        searchramlinealgen
            pop         bc
            pop         hl
            jr          nc,searchramlineal48k2
            inc         hl
            djnz        searchramlineal48k1
            ret

searchramlineal48k2:
            ld          (ramlineal2),a
            ld          (ramlineal1),a

searchramlineal48kend:
            ld          a,RAML32K
            ld          (ramlineal),a
            ret

; Analiza el slot pasado en A
; Con los de otro buffer
; HL buffer
; DE Contador
; Cy = 1 No encontrado

searchramlinealgen:
            ld          c,a
            ld          a,(de)
            ld          b,a

searchramlinealgen0:
            ld          a,(hl)
            cp          c
            ret         z
            inc         hl
            djnz   searchramlinealgen0
            scf
            ret
```

[LA PAGINA 0 Y LA PAGINA 3]

Como hemos visto en las rutinas, cuando hemos buscado la máxima memoria, hemos ignorado completamente la pagina 3. Esto tiene una explicación. Trabajando con la BIOS y el sistema MSX de "casa" (es decir no haciendo un sistema operativo propio que trabaje de otra manera), no tiene sentido cambiar la página 3 para nada. En ella se ubican todas las variables del sistema, muy necesarias para trabajar con la BIOS y el hecho de cambiarla, además de ser complejo, no ayudaría mucho por su complejidad a nuestro programa. Para cambiar la página 3 necesitariamos habilitar otra zona de memoria para tener la pila y variables. La pila normalmente se ubica en la página 3, que es donde está la memoria desde que arranca el MSX como hemos comentado. El hecho de cambiar esa página pasa obligatoriamente por mover la pila y almacenarla en otro sitio. No obstante, claro, se puede hacer.

Y luego la página 0. Esta si que es útil cambiarla. Pero tiene un problema. Así como para trabajar con la página 1 y 2 e ir cambiando el slot/subslot de las mismas podemos usar la rutina ENASLT de la BIOS, para cambiar la página 0 esto no es posible, debido a que

[] El código de la rutina ENASLT se encuentra precisamente en esa página, la 0. Aunque la rutina se apoya en la página 3 (en una zona del sistema de variables) para hacer el cambio, es capaz la mayoría de las veces de posicionarte el slot/subslot de la página 0, pero *luego es imposible volver a poner la BIOS otra vez u otro slot/subslot con ENASLT*. Es obvio puesto que al cambiar la página 0, que es donde estaba el código de ENASLT pues ya no tenemos acceso a él. Esto es uno de los pequeños fallos del sistema, ya que lo correcto hubiese sido habilitar en otra página (la 3 era la indicada) una porcion de código que hiciese esta faena. Pero... no es posible tampoco. No es posible porque para cambiar un slot expandido es necesario poner el slot en la página 3, cosa que se sale de este artículo y explicaremos en el siguiente.

[] ENASLT activa siempre las interrupciones. Lo cual es un peligro. Otro fallo del sistema. En lugar de habilitar una variable para que al salir de ENASLT activase o no la interrupciones según esa variable, por defecto las activa siempre. ¿Cuál es el problema? Pues que si cambiamos la página 0, estamos en IM1 (donde la interrupción salta a 038h) con ENASLT, se activan las interrupciones y se produce una interrupción y, NO tenemos preparado nada en 038H en el slot/subslot destino, lo que vamos a conseguir es un bonito cuelgue seguramente.

```
; *** VARS ***

rampage0:        equ      0E000h
rampage1:        equ      rampage0     +    1
rampage2:        equ      rampage1     +    1
rampage3:        equ      rampage2     +    1
ramcheck0:       equ      rampage3     +    1
ramcheck1:       equ      ramcheck0    +    1
ramcheck2:       equ      ramcheck1    +    1
ramcheck3:       equ      ramcheck2    +    1
ramtypus:        equ      ramcheck3    +    1
slotram:         equ      ramtypus     +    1

rambuffer0:      equ      slotram      +    1
rambuffer1:      equ      rambuffer0   +    16
rambuffer2:      equ      rambuffer1   +    16
rambuffer3:      equ      rambuffer2   +    16
ramcounter0:     equ      rambuffer3   +    16
ramcounter1:     equ      ramcounter0  +    1
ramcounter2:     equ      ramcounter1  +    1
ramcounter3:     equ      ramcounter2  +    1

thisslt:         equ      ramcounter3  +    1
ramlineal:       equ      thisslt      +    1
ramlineal0:      equ      ramlineal    +    1
ramlineal1:      equ      ramlineal0   +    1
ramlineal2:      equ      ramlineal1   +    1
ramlineal3:      equ      ramlineal2   +    1
```

Incluso las diskROMs de las cuales hemos hablado mucho lo hacen a mano. Es decir, no usan ENASLT, y cuando arrancan el MSXDOS cambian esta página para poner la memoria a mano. Pero, como hemos dicho, esto queda ahora un poco fuera del artículo, y lo completaremos en el siguiente donde nos centraremos en estas dos problemáticas páginas y como poder cambiarlas sin morir en el intento.

[¿Y PARA QUE VALE ENTONCES TODO ESTO?]

Pues para mucho. De momento ya podéis usar con toda libertad las rutinas y usar la RAM encontrada para páginas 1 y 2, mediante ENASLT. Desde luego es bastante, hasta el siguiente número. Pero como lo mejor es tener algo que demuestre su uso, aquí os dejo el link a una cosa que hace uso de todas estas rutinas, y que implementé para su prueba. Son tres juegos de casete pasados a ROM. Necesitan de un ordenador de 64k, esté como esté la memoria y funcionan en cualquier caso. Adiós pokes mágicos, adiós incompatibilidades. Aquí tenéis las ROMs junto con el código fuente del artículo, en dos versiones, para mapper ASCII8 y mapper KONAMI5:

http://callmsx.atlantes.org/8/64kmem.zip

[SOBRE MEMORY MAPPERS Y GESTORES DE MEMORIA]

Al principio del artículo hablamos de ellos y dijimos que no nos íbamos a centrar en esos asuntos por ahora. Hay mucha documentación de como trabajar con ellos, como averiguar su capacidad y como usar los gestores de memoria de MSXDOS2. Todo esto ha sido explicado varias veces por el maestro Konamiman, y creo que sobra repetirse. Aún así si tenéis cualquier duda ya sabéis donde encontrarme.

[NOTAS FINALES]

Si habéis sido capaces de llegar hasta aquí, seguro que os habréis hecho las mismas preguntas que me hice yo en su día. ¿Por qué todo esto no lo hace el sistema operativo o la BIOS? Pues no lo se. Es otra de las lagunas del MSX. Creo que no les hubiese costado mucho, ya que obligatoriamente busca la BIOS memoria, apuntar en 4 bytes (4 variables del sistema) el resultado que nos da la rutina <searchramnormal>. Esto hubiese evitado MUCHOS, MUCHISIMOS problemas de compatibilidad en en standard. Todavía no me explico porqué lo hicieron así.

Lo que sí espero es que toda esta explicación evite para siempre problemas de compatibilidad y despeje muchas dudas. A día de hoy, con toda la información en nuestra mano siguen apareciendo muchos programas incompatibles. Los slots/subslots y la memoria siguen siendo problemáticos para los nuevos programadores del sistema. Y ya no deberían de serlo. Si queréis hacer pruebas de forma fácil, no es necesario que llenéis vuestra casa de MSXs. Con el emulador openMSX es sencillísimo "construirse" cualquier extraña configuración de MSX, poniendo la memoria como queramos. De hecho yo me configuré un MSX de 48k y el sistema funciona perfectamente. Si, he dicho mil veces que es raro, pero no imposible. De hecho, de nuevo la diskROM ya comprueba que realmente exista memoria en página 0 y 1 para arrancar MSXDOS. Si el sistema obligase a pasar de 32k a 64k, podrían evitarse esa comprobación y sólo comprobar una de las dos páginas. Pero no, comprueban ambas lo cual quiere decir que el sismema permite este tipo de ordenador. En fin, lo dicho, aquí me tenéis para lo que queráis. En el siguiente número profundizaremos en el tema de cómo cambiar la página 0 sin morir en el intento, y de paso, cómo construir ROMS de 48k que va directamente unido.

¡¡¡Hasta el próximo número!!!



¿Te crees capaz de compatibilizar tu soft de 64k con éste?

Armando Pérez (Ramones)
e-mail : **majingazetto@gmail.com**
Fidonet: **2:343/107.20**



RC710 - RC711
Hyper Olympic / Track & Field 1 y 2
16Kb ROM
Konami 1984

Con la inminente celebración de los juegos olímpicos de Los Ángeles de 1984, Konami presentó Track & Field, una revolucionaria máquina recreativa inspirada en este gran encuentro y en la que las pruebas se superaban machacando literalmente el joystick. Más tarde dos cartuchos aparecerían para MSX: Track & Field 1 y 2 o Hyper Olympic 1 y 2.

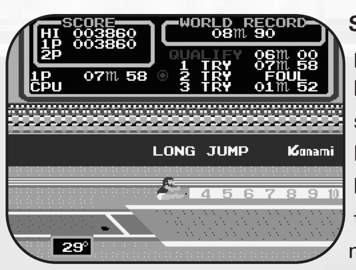
Los cartuchos Track & Field 1 y 2 engloban en total ocho de las diez disciplinas atléticas que forman el decatlón (quedando fuera el lanzamiento de disco y el salto con pértiga). En cada una de estas pruebas, el jugador debe conseguir superar la marca mínima establecida para la clasificación y que le permite continuar en el torneo; aparte de esta marca existe un récord mundial para cada una de las pruebas y nuestro verdadero objetivo es superarlo..

El juego repite una y otra vez la tanda de pruebas e incrementa la marca mínima de clasificación por lo que cada vez es más difícil mantenerse en el torneo.

Track & Field I



100 metros lisos. La primera de las pruebas es una rápida y corta carrera entre dos atletas. Hay que correr al máximo para superar a nuestro contrincante y a la vez obtener una buena marca por encima de la mínima de clasificación. Pero ojo, porque nuestro adversario apretará el ritmo cada vez que se vea superado. Para poder correr hay que ir pulsando las teclas izquierda y derecha alternativamente lo más rápido posible.



Salto de longitud. Una de las pruebas más conocidas en las que debemos correr y saltar lo más lejos posible. No podemos pisar la línea límite de salto ya que cometeremos una falta y perderemos el intento. Disponemos de tres intentos para superar la nota de clasificación. Los controles para correr son los mismos que en la anterior prueba pero hay que añadir la barra espaciadora para realizar el salto, que será de mayor o menor ángulo en función del tiempo que tengamos pulsada dicha tecla.



Lanzamiento de martillo. La prueba consiste en lanzar un objeto pesado lo más lejos posible. Para ello el atleta gira rápidamente sobre sí mismo para ganar toda la fuerza centrífuga posible y así conseguir lanzar el peso con mayor fuerza. Para realizar el giro hay que pulsar las teclas izquierda y derecha alternativamente, lo más rápido posible; en el momento que nos encontremos cerca del límite de lanzamiento y nuestro personaje esté mirando hacia la izquierda pulsaremos la barra espaciadora para lanzar la pesada bola. Dependiendo del tiempo que dejemos pulsada la tecla, el ángulo de lanzamiento será mayor o menor, como sucede en la prueba anterior.

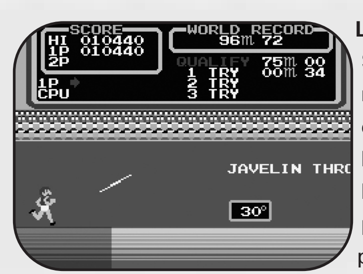


400 metros lisos. De la misma manera que hacíamos en la primera prueba, ahora hay que correr 400 metros, por lo que estamos obligados a administrar mejor las fuerzas, reservando siempre una parte de ellas para el último moemento, y así conseguir llegar a la meta dentro de el tiempo de clasificación y obtener una buena nota.

Track & Field II



110 metros vallas. Muy conocida también; en esta prueba tenemos que correr una distancia relativamente corta pero añadiendo la dificultad de que tenemos que saltar las vallas. Como en las otras pruebas, corremos con los cursores izquierda/derecha y saltamos con la barra espaciadora. Hay que calcular muy bien el momento del salto y evitar tropezarnos ya que perderemos mucho tiempo.



Lanzamiento de javalina. Similar al lanzamiento de martillo pero de más sencilla ejecución. Hay que correr hasta llegar a la línea de lanzamiento en donde, sin pisar ésta, tenemos que pulsar espacio tanto tiempo como ángulo queramos obtener y soltar para lanzar la jabalina.



Salto de altura. Esta es una de las pruebas más difíciles de ejecutar ya que hay poco espacio para correr y a la vez hay que ser muy preciso en el salto. Como siempre, con los cursores tenemos que correr al máximo y al llegar al punto de salto (un poco antes de la barra) tenemos que pulsar la barra espaciadora durante unas décimas de segundo para ganar algo de altura y luego soltarla para que nuestro atleta se estire y pueda pasar por encima de la barra sin tocarla.



1500 metros. La última prueba es la más dura de todas. Hay que correr un kilómetro y medio por lo que el cansancio se hará notar enseguida. Hay que controlar muy bien el ritmo y dejar una reserva de energia para esprintar en los últimos metros. El contrincante dará un arreón cada vez que le dejemos en el extremo izquierdo de la pantalla, es entonces cuando deberemos demostrar nuestro sprint.

Secuelas

Konami supo aprovechar este título para vivir de rentas en otros sistemas. En 1987 hacía el salto a la consola de 8 bits de Nintendo, "Track'n Field" con un total de ocho pruebas incluyendo alguna extra de los Hyper Sports como lo era el tiro con arco. Prácticamente idéntico a la versión MSX. Lo fuerte es que lo volvieron a vender en el año 1991 como "Track'n Field in Barcelona" para aprovechar el tirón. Lamentable. La segunda parte de 1988 "Track'n Field 2" o "Konamic Sports in Seoul" sí que supuso un cambio en todo el concepto, pruebas y gráficos. Después verían versiones casi todas las consolas.



El cambio poligonal llega en 1996 para la Playstation con "International Track 'n Field" buscando el realismo a pesar de los escasos polígonos de la 32 bits de Sony. Hay que fijarse siempre en las fechas de lanzamiento que coinciden descaradamente con la celebración de los juegos olímpicos. Para las de Sydney 2000 llegó otra secuela llamada "ESPN International Track'n Field" ya para las consolas de 128 bits (PS2 y Dreamcast). Y no ha vuelto a salir nada más hasta este año en el que atacan con un juego casi parodia para la Nintendo DS llamado "New International Track'n Field". En él podremos ver desfilar a personajes ilustres de Konami... ¿no os suena demasiado esto?



Soft Amateur

Una de las secciones que nos causa más simpatía, porque encontramos proyectos interesantes, desde pequeños programas de iniciación al videojuego hasta auténticas demos donde se experimenta y se explota todo el potencial de nuestra querida máquina. En esta ocasión comenzamos con la demo más bestia vista jamás en un MSX y le sucederán varios programitas llegados del país del sol naciente, con más o menos fortuna en su realización. Esperamos que todavía sean una sorpresa para algunos usuarios.

UTOPIA

Casa : DVIK & JOYREX 2007
Música : PSG
Tipo : Demo
Sistema : MSX1 (DSK y ROM)

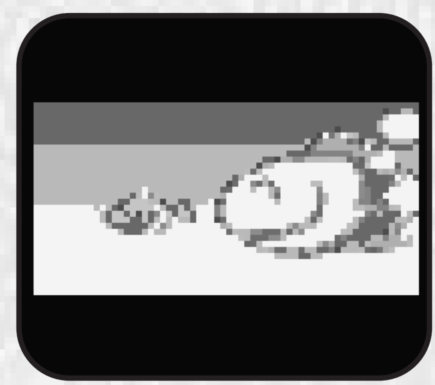
Segunda demo de esta pareja de programadores que buscan sacar el máximo partido a los ordenadores de primera generación, y ¡vaya si lo consiguen! Por suerte para nosotros nos dieron el privilegio de presentarla en primicia en la RU de Barcelona antes de subirla al Resource Center.



El efecto de la presentación del logotipo es fascinante, y todo se resuelve jugando con la pantalla negra superpuesta sobre el fondo blanco, aunque parezca al revés. La tipografía es contundente y el borrado final cumple.

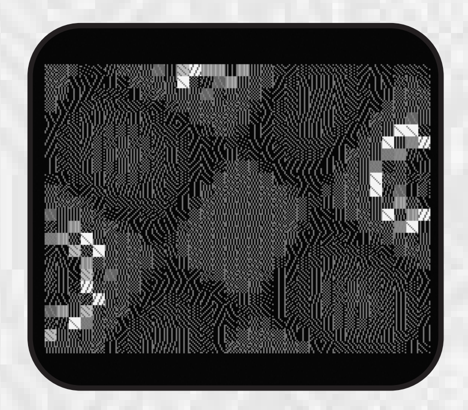
Aparecen los créditos de presentación, el nombre real de los verdaderos creadores, los ya conocidos Vincent Van Dam y Daniel Vik. El sonido que acompaña y sonará en las demás ocasiones que cargue la versión disco es un guiño al Loading de la archiconocida demo *Unknown Reality*. Vamos que han clavado el sonido si es que no lo han tomado "prestado" directamente.

Tras esto suena una música estilo techno, sonido típico de demo ochentera y se unen horizontalmente una trama de cuadros tipo tablero de ajedrez tanto por la izquierda como por la derecha para unirse y cerrar la pantalla. Entonces se crea un scroll vertical y aparecen textos con una tipografía grande y muy cuidada que nos narra que ya hace un año desde su primer lanzamiento conjunto, *MSX Unleashed*. Nos dicen que nos agarremos fuerte que empieza la demo y desaparecen los cuadros uno a uno que formaban el tablero hasta quedar la pantalla en negro.

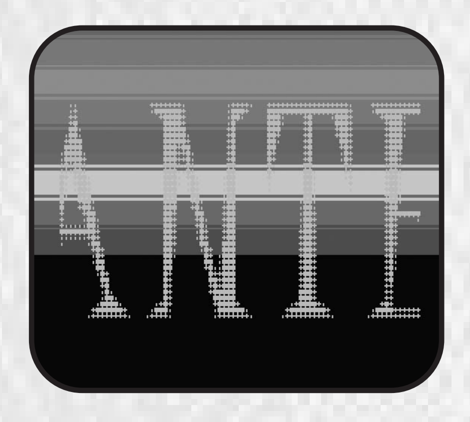


A continuación vemos una animación llamada *Nature for dummies* dentro de una franja panorámica. Es algo que nos puede recordar a los Evas. Pero sabemos que esto está hecho de otra pasta. Es una cadena alimentaria de peces. Le acompaña un tema musical de estilo electrónico experimental.

Ahora le toca el turno a ver las capacidades de la máquina en cuanto a zoom y rotación. El motivo gráfico, el primero es abstracto y el segundo es un ojo que se multiplica cada vez que se aleja el zoom. Es francamente espectacular, ya pudimos ver algo parecido en su primera demo. Bastante más complejo que lo visto en Unknown Reality y eso que aquella demo era para computadoras de segunda generación.



Tras otra música caótica aparece un texto en scroll lateral con una tipografía muy grande y con efecto de tramado. Además éste consigue ensancharse y encogerse mientras aparecen un par de franjas, una roja y otra azul degradadas, superponiéndose entre ellas por detrás del texto. Espectacular.



Luego aparecen un par de imágenes fotográficas en screen 0 con una burrada de colores. Verlo para creerlo, como en un efecto de entrelazado. Sigue



sonando música “rara”.

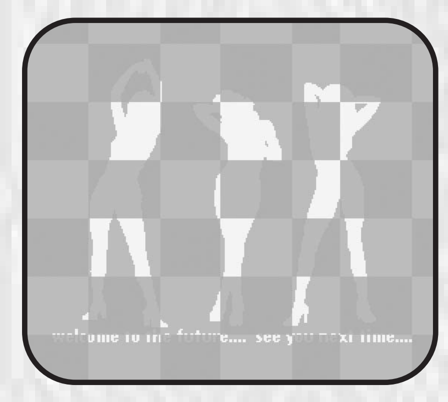
Después viene un efecto muy guapo de rotaciones es screen 2 que quiere provocar la sensación en el espectador que estamos avanzando como por una especie de túnel rojo, con el texto Old School simulando estar en una capa por encima del efecto matemático. Además consigue un efecto de romper la perspectiva frontal pues el espectador rota sobre sí mismo por el túnel. Aparte, tanto la paleta como la textura del gráfico también cambia. Una auténtica pasada.



Luego una línea discontínua de puntos se deforma haciendo curvas sinuosas "bailando" al compás de la



música. Mientras, aparece y desaparece otra imagen fotográfica de fondo, en screen 1. Ésta, además, va cambiando de color el área vacía o ausencia de pixelación negra.



Por último una pantalla en screen 0. Scroll vertical de una tabla de ajedrez y de fondo inmóvil las siluetas de 3 chicas.

LIST PERIOD

Casa : Suzumizaki-Kimitaka, 1996
Música : FM o MIDI
Tipo : Matamarcianos
Sistema : MSX2 (Turbo R)



Estamos ante una demo de un curioso matamarcianos, que si bien es válido para los ordenadores de segunda generación es altamente recomendable para MSX turboR, ya que la velocidad del procesador R800 permite una suavidad en el control del juego más que notable.

Tenemos la duda si el juego completo existe, parece que sí, al precio de 600 yens (si es que todavía lo venden después de tantos años). El caso es que nos han dejado una demo jugable para echarle un vistazo. Si os habéis fijado bien, los autores son los mismos del espectacular pero breve (¿pasó de ser una meritoria demo?) Metorite Kiss.



La pantalla de presentación no es nada del otro mundo. El juego en sí tampoco, pero sí es original, en el enorme tamaño de los sprites. Choca este tamaño y más en un shooter. Parece que no vamos a tener espacio suficiente para maniobrar con nuestra exagerada nave. Pero verlo para creerlo. Claro, estamos hablando con el juego cargado en un TurboR. Es una delicia en cuanto a suavidad en la respuesta al teclado. Y vamos disparando a todo lo que se te viene encima que no es poco.

Decir que cada vez se hace más difícil, pues las hordas de enemigos que vienen en formación también te disparan. Como siempre enemigo de fin de fase que no respetará un hueco en la pantalla. No estás seguro en ningún rincón. Decir que tiene cinco niveles de dificultad para elegir.

A primera vista parece que el escenario de fondo siempre van a ser estrellas... pero no, va variando con nuevos elementos. Aunque eso sí, no es ninguna maravilla. Son gráficos simplones y tan vastos como los sprites. Eso sí, no hay confusión entre primer plano y fondo.

Soft Amateur

Si no disponemos de Midi el sonido éste será FM. No hay música hasta que comienza la partida. Efectos de disparos y explosiones en PSG y melodía FM. El tema tampoco es gran cosa; nos llama más la atención cuando nos matan y sale el mensaje de continuar con F5 pues la melodía nos recuerda mucho a un tema de Compile (seguro que también os suena). En el BlueMSX no nos funcionó el Midi, al contrario que con Meteorite Kiss.

JASMINE STAR

Casa : Soft Mikeen 1994
Música : PSG
Tipo : Matamarcianos
Sistema : MSX2



Pequeño programa en el que nos enfrentamos a un típico jefe final, en este caso un dragón azul. Simplemente es eso, derrotarlo, y en cuanto lo hagamos pasaremos de pantalla y volveremos a enfrentarnos a él, pero cada vez utilizará armas más sofisticadas para acabar con nosotros.



A destacar el scroll horizontal mientras luchamos, el cual notamos por el movimiento de fondo de los puntos blancos que simulan ser estrellas. Eso sí, podemos ver gran cantidad de sprites en pantalla debido a los numerosos disparos entre uno y otro bando.

Una verdadera lástima que sólo haya quedado en eso, porque se percibe una calidad más que notable. Imaginarse un juego completo visto lo visto nos hace soñar.

http://kakomiki.hp.infoseek.co.jp/msx/msx5.html

PHI (FIRE)

Casa : Hishidama's Soft 2001
Música : NO
Tipo : Puzzle
Sistema : MSX2



Sorprendente programa de puzzle que nos llama la atención por la "reciente" fecha de creación para ser soft japonés.

Sí, es difícil encontrar soft del país del sol naciente después del año dos mil. Para los frikis del soft japonés aquí tenemos una pequeña joyita.

El concepto del juego es muy básico. Debemos llegar a la puerta, que se está quemando, apagando los fuegos que nos impiden el paso. Para ello la solución es mover los bloques de hielo para que se deshagan y de esta manera



apagar el elemento ígneo. La cosa se va complicando con tuberías tipo Mario Bros de donde salen bolas de fuego, socavones y enemigos varios. Con las teclas Z y X podemos crear más bloques de hielo cuando la ocasión lo requiera.

Podemos modificar la velocidad, al menos al jugarlo en un TurboR.

Un jueguecillo divertido, para pasar el rato, con gráficos llamativos pese a no ser espectaculares. A destacar el efecto de fundido cuando nos pasamos una pantalla. La gran pega es no tener música ni efectos, lo que le resta mucha vidilla.

http://www.ne.jp/asahi/hishidama/home/soft/phi/phimsx.html

SEIHAIDEN

Casa : Soft Mikeen
Música : MSX MUSIC
Tipo : Puzzle
Sistema : MSX2

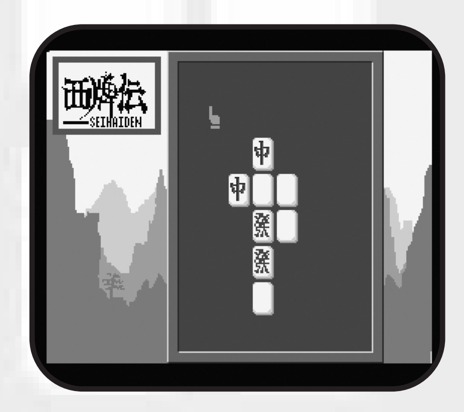


Otro programita japonés en el que debemos conseguir quedarnos únicamente con una ficha de cada tipo, por lo que deberemos tener mucho cuidado hacia dónde movemos las fichas. Si fallamos podemos pulsar Graph y volver a reintentarlo.



El grafismo es correcto. A destacar la ilustración inicial en la pantalla del menú, un dibujo manga de una chica tipo Chun Li bien trabajado en Screen 5.

La pantalla de juego no deja de ser una simple pizarra sobre un paisaje montañoso idílico chino donde se reparten las fichas que debemos mover.



La música se agradece que utilice el FM, elaborada con el programa Musica. Hay varios temas pero no son muy espectaculares. Algunas son poco melódicas, se limitan a mantener el ritmo con el bajo y la batería.

Un detalle curioso que nos sorprende gratamente es la inclusión de Passwords para no comenzar de cero la partida.

Un juego del montón, pero no estará de más probarlo.

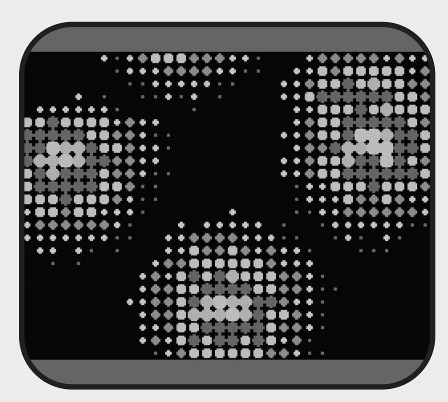
http://kakomiki.hp.infoseek.co.jp/msx/msx0.html

PLANET OF THE EPAS

Casa : Traktor, 2007
Música : PSG
Tipo : Demo
Sistema : MSX1



Esta demo participó en el concurso de demoscena Breakpoint 2007 en la categoría de consolas, compitiendo con máquinas muy potentes y obteniendo un digno undécimo puesto de un total de diecisiete participantes.



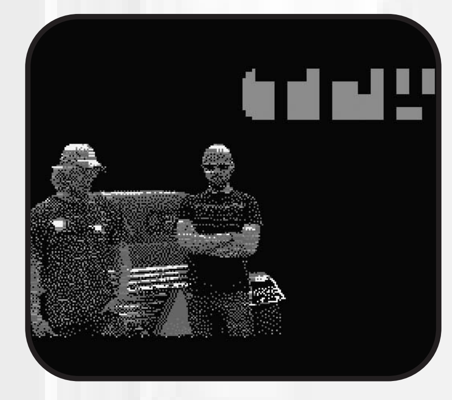
Se puede calificar en resumidas cuentas como una demo breve pero intensa. En poco más de un minuto nos sorprenden con multitud de efectos muy logrados y espectaculares. Se supone que esta manera de hacer busca un objetivo claro, llamar la atención en el concurso.



Nos encontramos los típicos zooms, rotaciones y deformaciones del plano con un aspecto visual muy llamativo. También vemos torsión y estiramiento de elementos al más puro estilo de un muelle; son efectos que ya hemos visto en otras demos pero cambiando el grafismo.

Luego vemos rotaciones matemáticas sobre varios ejes a velocidades increíbles. Y terminamos con un plano de rejilla al que se desplaza y se retuerce ante nosotros. Se despiden con una digitalización de los creadores en screen 2.

Y todo ello acompañado por un tema musical en PSG bastante loco, entre techno y makinero, que sustituye a veces la melodía por efectos de sonido típicos de los videojuegos. Funciona muy bien el tema machacón con la velocidad general que ofrece la demo.



http://www.msx.org/Planet-of-the-Epas.newspost4236.html

USAS

INPUT KEYWORD
MOHENJO DARO

Contraseñas

Para introducir las contraseñas hay que pulsar la tecla **"CTRL"** en la pantalla del título; una vez escrita hay que confirmar con **"RETURN"**.

Nivel 2
JUBA RUINS

Nivel 3
HARAPPA RUINS

Nivel 4
GANDHARA RUINS

Nivel 5
MOHENJO DARO

Si nos equivocamos escribiendo el código no podremos rectificar y nos lo dará como erróneo.

Combinaciones

Todos los cartuchos a combinar se insertan en el Slot 2.

Metal Gear

El daño que nos pueda producir el enemigo será la mitad del habitual.

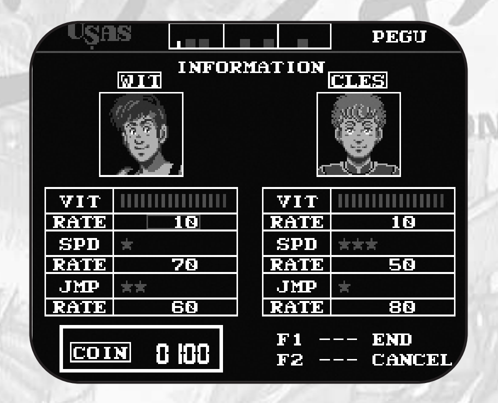
Gradius 2

Al finalizar la partida se nos dará la opción de continuar.

GAME OVER
F5 CONTINUE

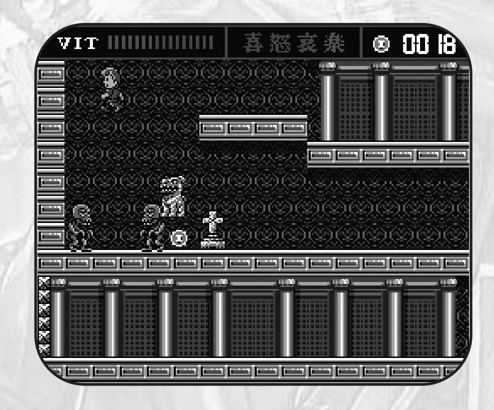
The maze of Galious

Empezamos a jugar con cien monedas extras.



F-1 Spirit

Nuestros personajes siempre dispondrán de las cualidades que tienen cuando se encuentran de buen humor.



Cles podrá realizar un salto doble en el aire.
Vit podrá caminar por el aire en vez de caer al vacío, cuando caminemos más allá del borde de una plataforma.

TWIN BEE



Equipo al máximo

Para empezar la partida con el escudo y los dos options hay que mantener pulsada la tecla **"Z"** y comenzar la partida pulsando la tecla **"M"**.



Este truco solamente funciona en el modo de un jugador.

Ambulancia

Para que la ambulancia siempre haga aparición cuando perdamos los brazos de nuestra nave hay que mantener pulsada la tecla **"T"** y empezar la partida pulsando la tecla **"M"**.



Barrera resistente

Si mantenemos pulsada la tecla **"W"** y pulsamos la tecla **"M"** para empezar la partida, cuando obtengamos la barrera, ésta durará el doble.

VENOM STRIKES BACK

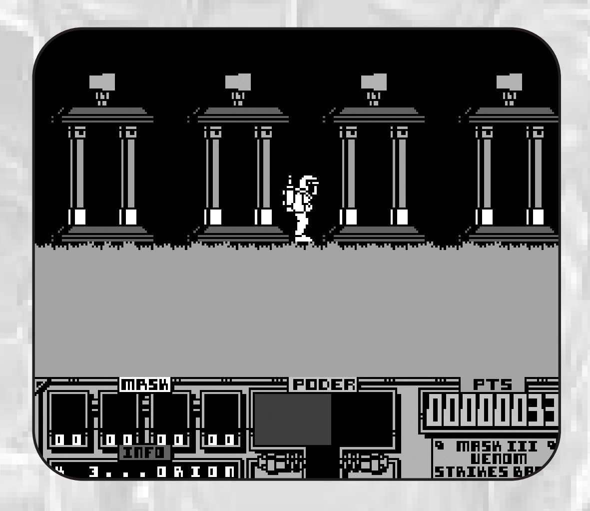


Contraseñas

Para introducir las claves hay que pulsar la tecla **"P"** en la pantalla de título. Cada clave abre uno te los teletransportadores que hay en el juego.

Claves de los teletransportadores

Nivel 1 MAYHEM
Nivel 2 TRANSMOGRIFY
Nivel 3 VALKYR
Nivel 4 PETALS OF DOOM



Solamente podremos abrir el teletransportador al nivel 4 si antes hemos abierto los tres primeros.

Una vez introducidas las claves, empezamos la partida. Encontraremos los teletransportadores caminando varias pantallas hacia la derecha.

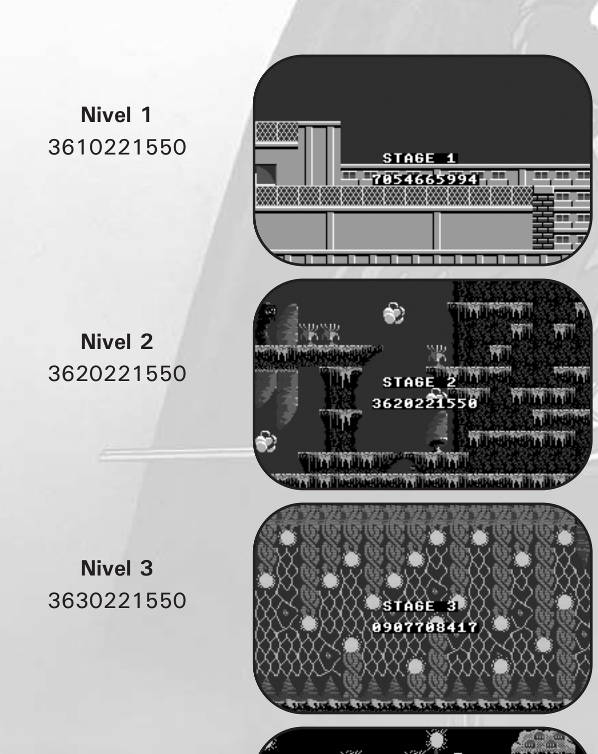
Para acceder al nivel deseado debemos colocarnos bajo su teletransportador y agacharnos (por defecto la tecla **"L"**).

VALIS



Contraseñas

Para que podamos introducir las claves que nos permiten acceder a los diferentes nives hay que pulsar la tecla **"RETURN"** cuando empiece la demostración del juego, donde se nos muestra uno de los niveles. Para introducir los dígitos de la clave se utilizan los cursores arriba y abajo.

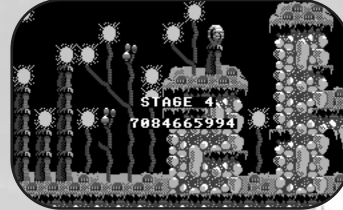


Nivel 1
3610221550

Nivel 2
3620221550

Nivel 3
3630221550

Nivel 4
3640221550



Nivel 5
3650221550

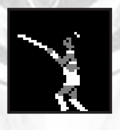


Funcionamiento de las claves

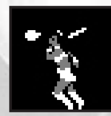
Las claves son generadas mediante un simple sistema de diez dígitos que responde a esta estructura:

3 w s ? a 2 1 c b k

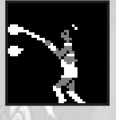
w - Tipo de arma



0 - Sable



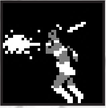
1 - Bola



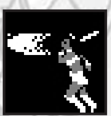
2 - Doble bola



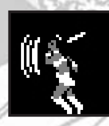
3 - Triple bola



4 - Fuego



5 - Super Fuego



6 - Onda

s - Nivel
Valores entre **0** y **5** dependiendo del nivel al que queremos acceder.

? - Función desconocida
Utilizar cualquier número.

k - Número llave
Para generar claves equivalentes. Dejar el valor a **0**.

abc - Nivel de vitalidad

"a" para establecer las centenas, **"b"** para las decenas y **"c"** para las unidades. El máximo de vitalidad se encuentra en **255**.

VALIS II



Test musical

Para acceder al menú secreto que permite escuchar las 33 melodías del juego hay que pulsar la tecla **"F-5"** en el momento que aparezca el título del juego. Una vez hecho esto aparecerá un mensaje solicitando que insertemos el disco **C**. Inserta el disco y pulsa **"RETURN"**. Accedemos al menú secreto en el cual Yuko nos invita a tomar una copa mientras seleccionamos las melodías con los cursores arriba y abajo, y las escuchamos con la barra espaciadora.
Si pulsamos **"RETURN"** podemos detener la melodía con un efecto de desvanecimiento o *fade out*.



WARROID



Menú de ajustes

Cuando aparezca la pantalla del título pulsar la tecla **"ESC"** y aparecerá el menú secreto.
Con los cursores podemos variar algunos parámetros como la altura del salto, la velocidad de movimiento o el color de los personajes y sus equipos, entre otros.
Para poder escoger el nivel en el que queremos empezar hay que mantener las teclas **"SELECT"** y **"BACKSPACE"** pulsadas mientras con **"INS"** y **"DEL"** subimos o bajamos el nivel, respectivamente.



MONTANA JOHN

Passwords

LEVEL2: *DRN2TB7MZV3P*	LEVEL 8: *AI6LEJ7WY4QM*
LEVEL3: *DRN6HJ7MZZ3T*	LEVEL 9: *NMCXWWL2L6Z5*
LEVEL4: *DSU1HJ7XYWQN*	LEVEL10: *NMCLQB7MTUZ8*
LEVEL5: *DSX6XWL2NB3P*	LEVEL11: *AI06S2UK4XDW*
LEVEL6: *DRPLRWLH3F3P*	LEVEL12: *AHCTAJ7XDYWY*
LEVEL7: *AHIIRPGNZR3P*	LEVEL13: *AI7ISB7XGTE2*

Los niveles de sangre/exp son mejorables pero permiten acabar cualquier nivel. El último permite acceder a todos los niveles previos, pero con "poco" por hacer...



WONDER BOY

Saltar de nivel

Para pasar directamente al nivel siguiente hay que pausar el juego con **"ESC"** y luego pulsar simultáneamente las teclas **"Z"**, **"W"**, **"S"**, **"X"**, **"E"**, **"D"**, **"C"**, **"R"** y **"F"**.

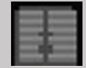
Cómo pasar ...



Ya no tenemos excusa para no llegar al final de este juego de Karoshi. Mirando las pantallas veremos todos los pasos que debemos dar para llegar con éxito a la salida. Seguimos el orden de los números (color gris para el arma y color negro para el enemigo). Para mayor comodidad también disponemos de los passwords para comenzar la partida en un punto más avanzado. Así que le damos las gracias a Antonio Robledo por facilitarnos la confección de esta estupenda guía.

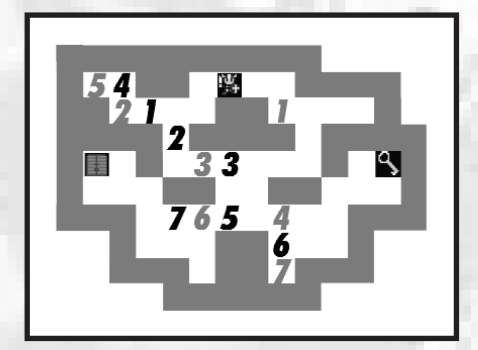
Leyenda

 Inicio

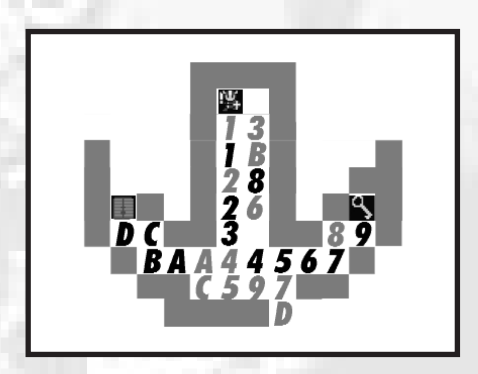
 Llave

Salida

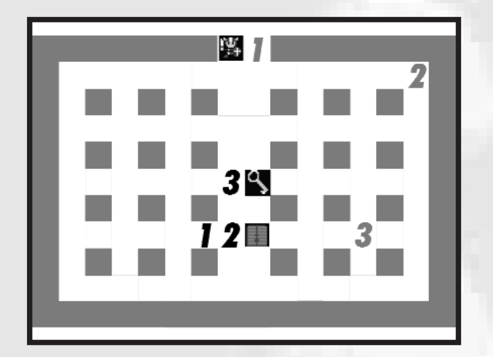
01



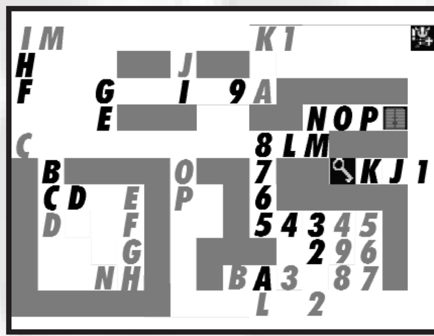
02



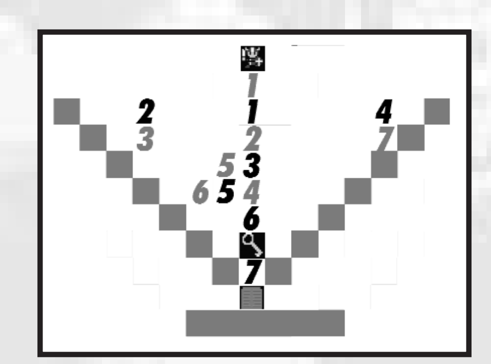
03



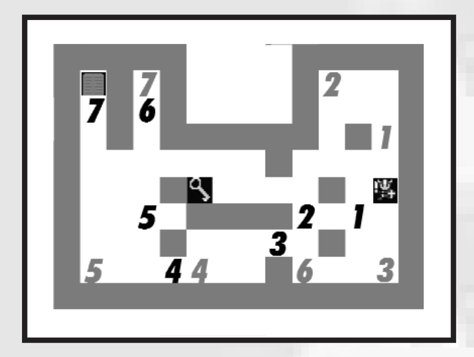
04



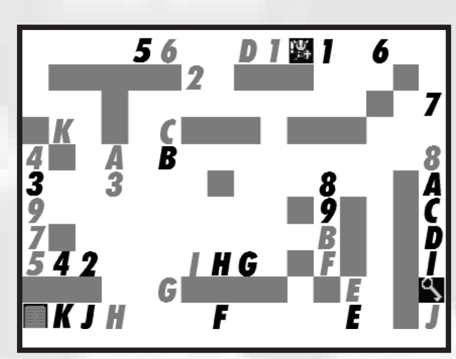
05



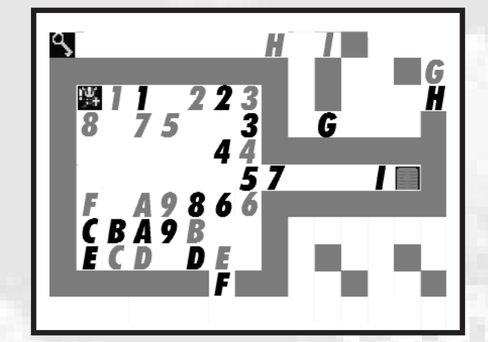
06 GENESIS



07



08



09

10

11 PSALMS

12

13

14

15

16 ROMANS

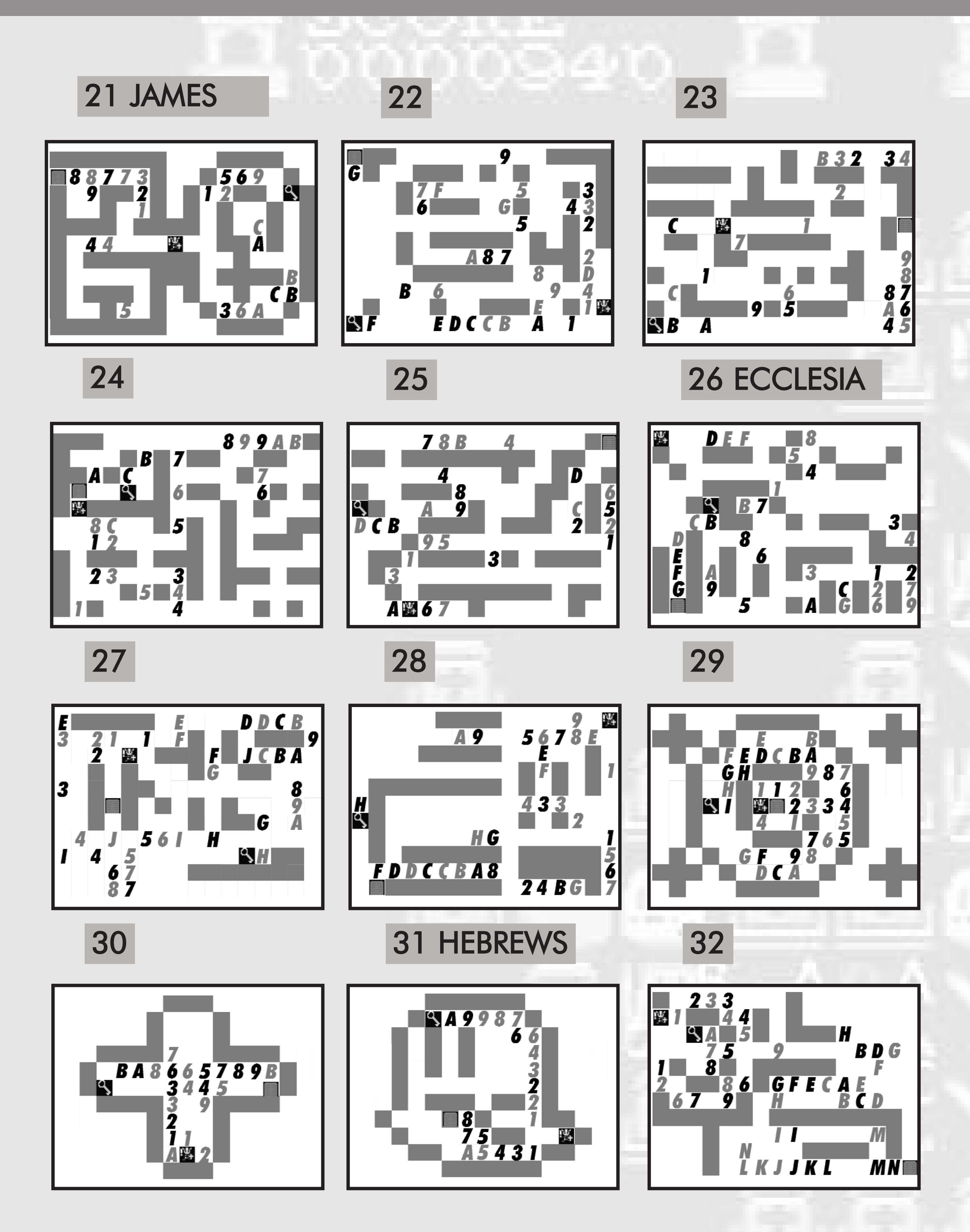
17

18

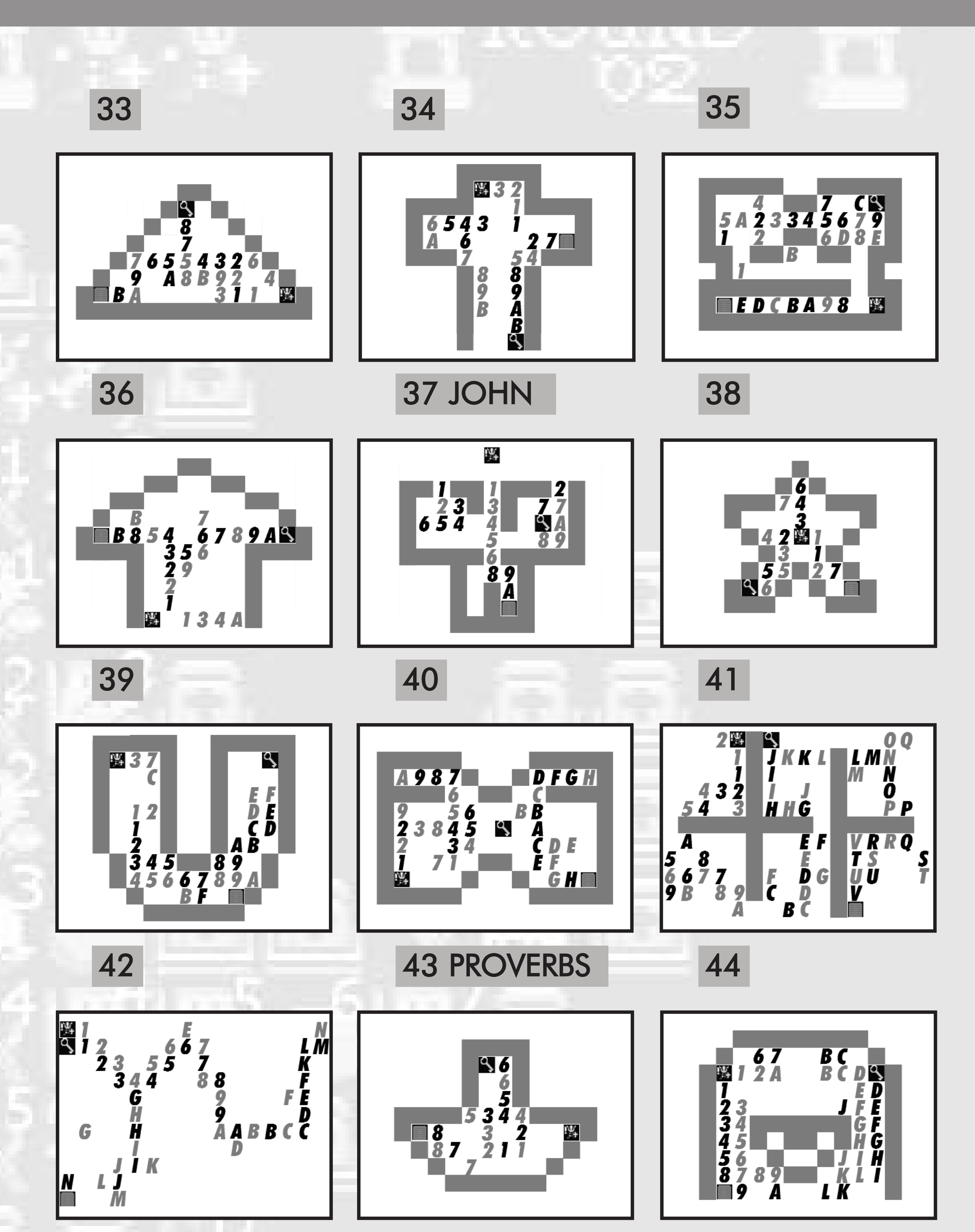
19

20

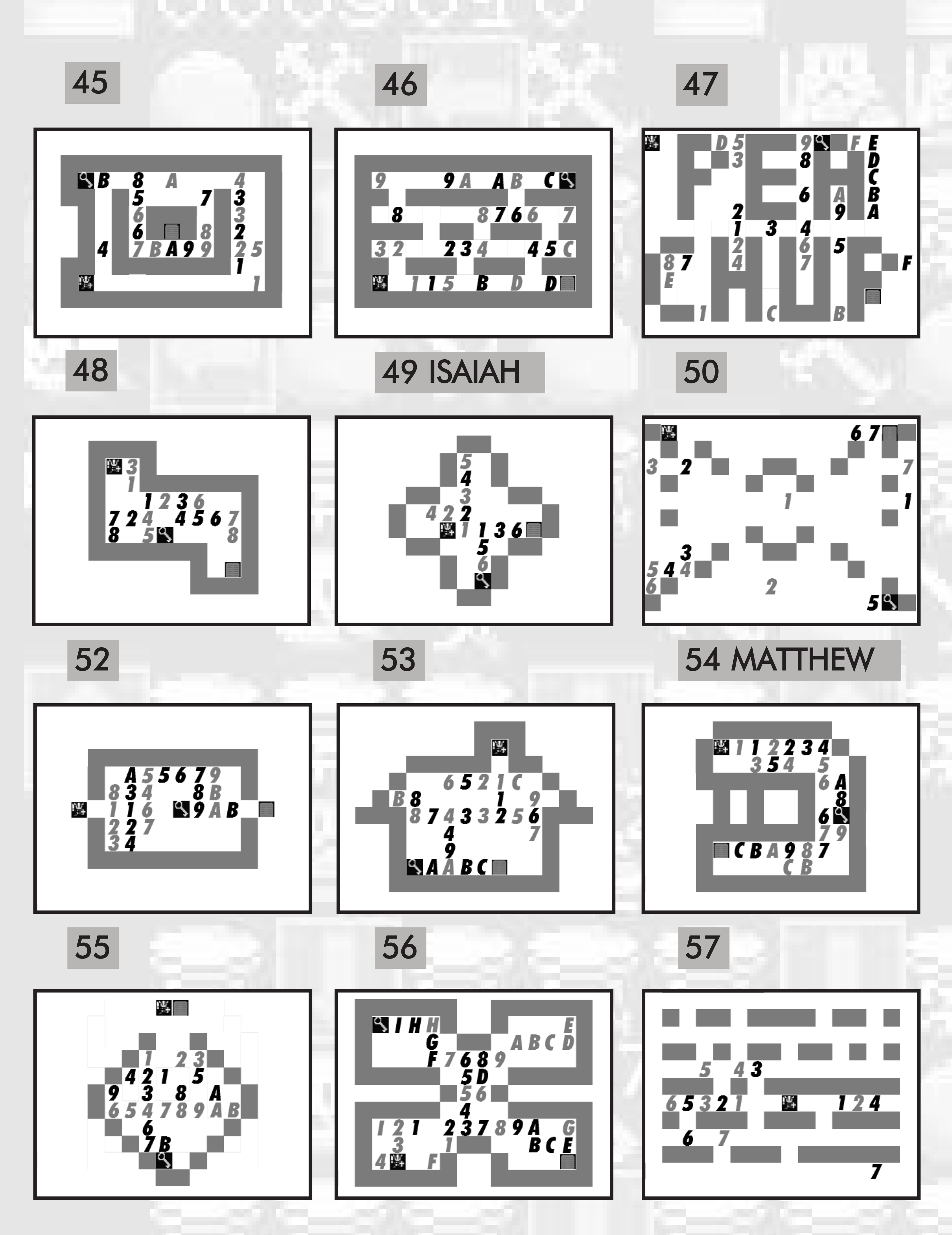
Cómo pasar ...



Griel's Quest



Cómo pasar ...



SNAKE IT!

Snake It! es un juego de puzzles programado por Ronald Pieket y publicado por EagleSoft en 1986. Está basado en el archiconocido juego de la serpiente donde tomamos el papel de una simpática serpiente (siempre está sonriendo) que tiene la misión de alimentarse dentro de los laberintos donde vive.

Lo cierto es que sería interesante conocer los motivos por los cuales esta serpiente, cuya raza crece de forma espectacular, ha quedado encerrada en esos laberintos en lugar de disfrutar del sol y las nubes en campo abierto; me temo que es algo que nunca sabremos a ciencia cierta.



Distribución de la pantalla

Un punto original del juego es que cada nivel tiene los marcadores situados en diferentes posiciones. Así, por ejemplo en el primero están situados a la izquierda del laberinto, mientras que en el segundo están encima.

Encontraremos cuatro marcadores diferentes:

-*Score*: puntuación obtenida
-*Bonus*: bonus que obtendremos al finalizar el nivel
-*Lives*: vidas que nos quedan
-*Round*: nivel en el que nos encontramos

El marcador de bonus se divide, a su vez, en bonus y multiplicador. Al comenzar cada nivel tendremos 1000 puntos de bonus multiplicados por uno. Este número irá decreciendo de 10 en 10 hasta llegar a cero. Sin embargo no hay que asustarse por esto, ya que es un simple marcador de bonus, es decir, no perderemos una vida al llegar a cero, simplemente no obtendremos ninguna bonificación. Así que el primer consejo es tomarse las cosas con calma.

¡A comer!

A diferencia de las versiones tradicionales del juego, aquí la serpiente no se mueve de forma constante, ya que en esta ocasión estamos ante un juego en el que la estrategia juega un papel fundamental, pues no toda la comida que encontremos tendrá los mismos efectos.

Las diferentes plantas que nos encontraremos en los diferentes niveles son:

-*Tréboles*: de color verde (20 puntos)
-*Flores*: de color blanco (30 puntos)
-*Setas*: de color rojo (venenosas)

Además, cada vez que comamos un trébol, una flor blanca aparecerá en otro punto de la pantalla y, por si esto no bastase, cada vez que nos comamos una flor blanca aparecerá una seta venenosa en otro lugar. Sin embargo esto no sucede en todos los niveles, ya que en algunos (por ejemplo el sexto) no aparecen setas al comernos las flores blancas.

Visto lo visto no es muy recomendable empezar a comer plantas de forma indiscriminada, ya que podemos quedarnos encerrados, en cuyo caso deberíamos pulsar la tecla ESC para comenzar el nivel desde el principio perdiendo, eso sí, una vida. El segundo consejo es comer primero los tréboles y luego las flores blancas.

Pero no todo son dificultades, también tenemos una ayuda en forma de ítems que irán moviéndose, más o menos al azar, por todo el laberinto:

-*Corazones:* aumentan en 500 el contador de bonus
-*Multiplicadores*: bolas con un número en su interior para multiplicar los bonus
-*Letra E*: si nos la zampamos obtendremos una vida extra

Niveles

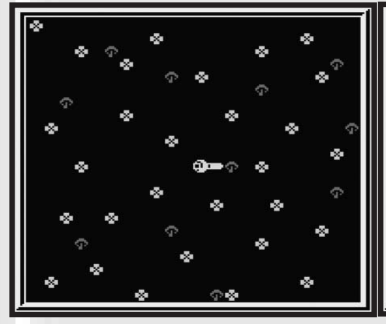
Este juego tiene un total de once niveles diferentes. A partir del duodécimo nivel se vuelven a repetir indefinidamente los niveles del seis al once. Esto es, el duodécimo nivel es igual que el sexto, el decimotercero igual que el séptimo y así sucesivamente.

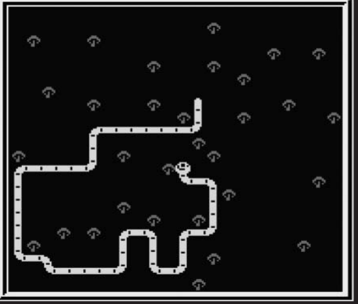
Cómo pasar ...

Para aquellos que tengan curiosidad sobre el juego os recomiendo que antes de descubrir como terminar cada nivel intentéis terminarlos por vosotros mismos. Para quienes sean impacientes o ya lo hayan intentado veamos cómo solucionar cada uno de los once niveles del juego:

Nivel 1

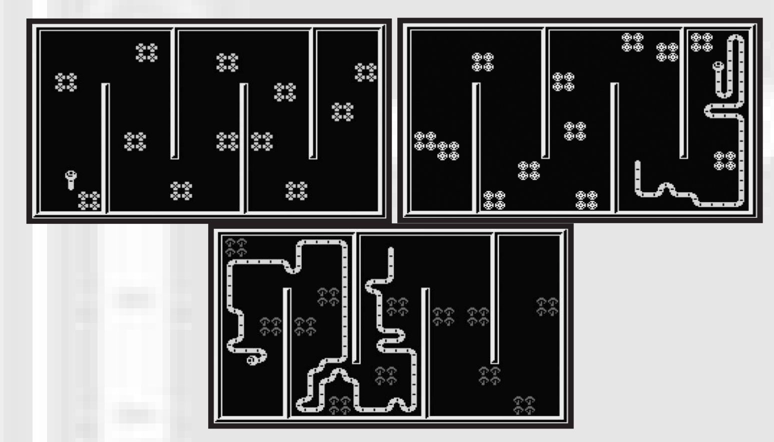
En la imagen (1) podemos ver la situación inicial del primer nivel. Al principio sólo veremos tréboles y setas (en ningún nivel aparecen flores blancas al comienzo). Este nivel es sencillo, ya que aplicar el primero de los dos consejos básicos (tranquilidad) es suficiente para terminarlo(2).





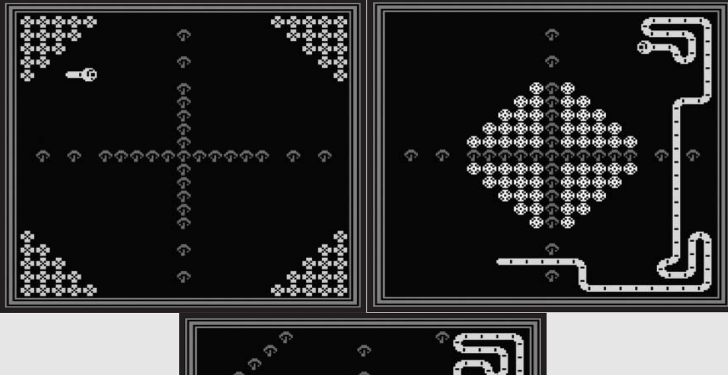
Nivel 2

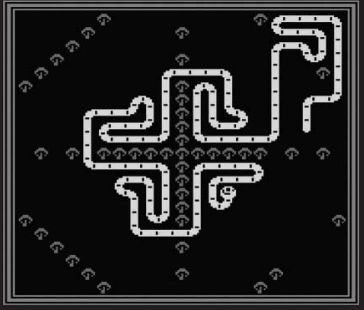
En esta ocasión el laberinto se complica levemente. Aparecen unas paredes que delimitan cinco zonas distribuidas en zigzag (1). En esta ocasión sólo hay tréboles, que deberemos devorar en su totalidad (2) para, a continuación, deshacer el camino y acabar con todas las flores blancas que hayan aparecido (3). Si no lo hacemos así, es probable que aparezca alguna seta roja cortándonos el paso.



Nivel 3

Volvemos a un nivel sin divisiones intermedias en el que veremos (1) cuatro zonas de tréboles situadas en las esquinas y una cruz de setas en el centro. La estrategia básica también es válida en este nivel: primero comemos los tréboles (2) y luego nos comemos las flores que aparecen en el centro (3). Si no lo hiciéramos así no podríamos comernos los tréboles porque las zonas donde están éstos se cierran al aparecer las setas.

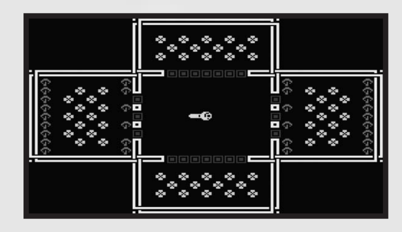




Nivel 4

Aquí las cosas se complican un poco y entran en juego unos bloques azules que pueden ser empujados por la serpiente siempre y cuando no haya ningún elemento detrás del bloque.

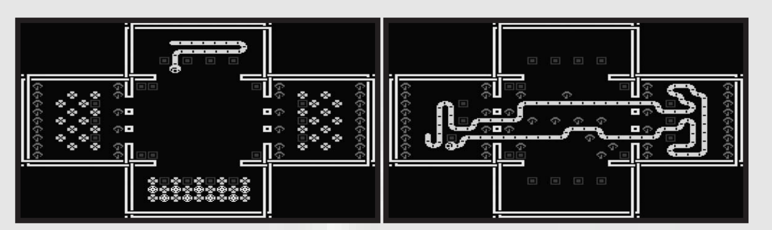
Este nivel en forma de cruz (1) se divide en cinco zonas separadas por bloques azules. En las cuatro zonas en de la cruz hay tréboles y setas (ojo con ellas) y una zona central donde aparecerán más setas cuando comamos las flores blancas.



Lo que hay que hacer es abrir paso a todas las zonas antes de empezar a comer nada. Si nos fijamos, las zonas superior e inferior están separadas de la central por una fila de siete bloques azules. Bien, lo que hay que hacer es empujar hacia el interior de estas zonas los cuatro bloques impares y meter en la zona central los pares. Las esquinas son buenos lugares para que no estorben.

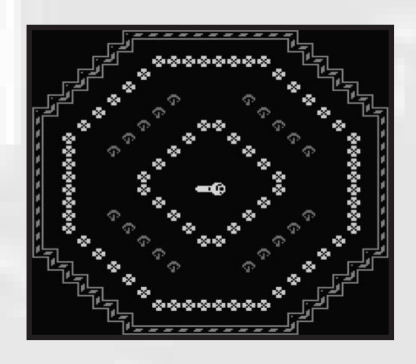
En las zonas derecha e izquierda hay tres bloques azules

que deberemos meter dentro de estas zonas. Una vez despejado el camino de entrada y salida a todas las zonas, podremos empezar a comer los tréboles (2) y las flores blancas hasta terminar el nivel (3).

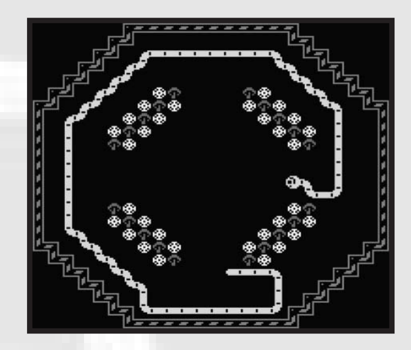


Nivel 5

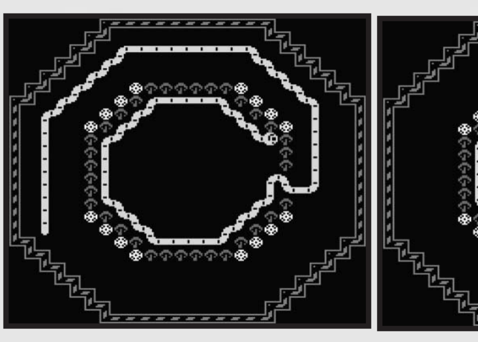
Nuevamente un nivel cuadrado sin división por zonas (1) con dos círculos (realmente son octógonos) de tréboles rodeando cuatro zonas de setas rojas. La trampa de este nivel aparentemente sencillo, estriba en el hecho de que, al comer las flores blancas, aparecerán las setas que faltan para cerrar el círculo de setas, dejándonos encerrados dentro o fuera del mismo y sin posibilidad de comernos todas las flores blancas.

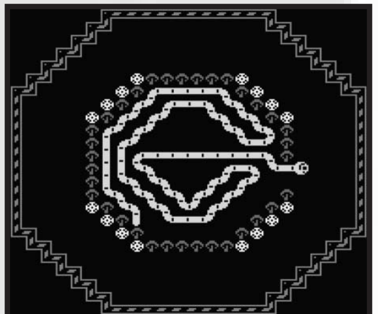


Para salvar la trampa, en primer lugar nos comeremos todos los tréboles, con lo que aparecerán todas las flores. A continuación, entraremos en el interior del círculo virtual de setas colocando el cuerpo de la serpiente de forma que algunas de estas setas no puedan aparecer (2).

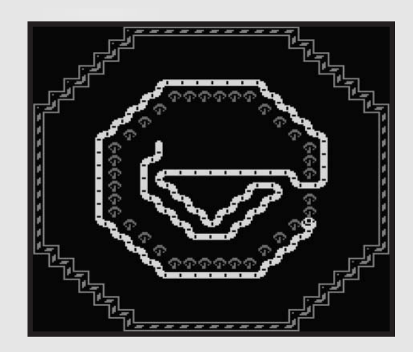


A medida que nos vamos comiendo las flores interiores veremos como el círculo de setas se va cerrando dejándonos, aparentemente, encerrados en su interior (3). Ahora aprovecharemos la flexibilidad de nuestra serpiente para cubrir con su cuerpo el interior del círculo de setas hasta que veamos que el hueco que dejamos queda libre (4).



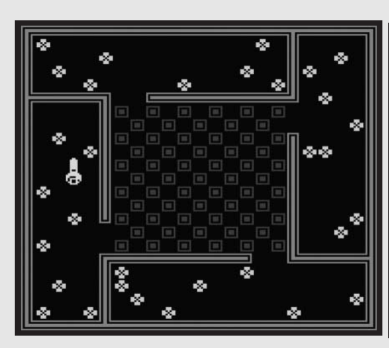


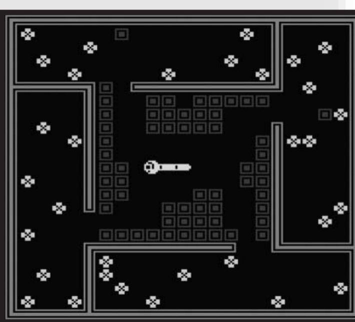
Sólo nos resta comernos las dieciséis flores blancas del exterior y habremos completado el nivel (5).



Nivel 6

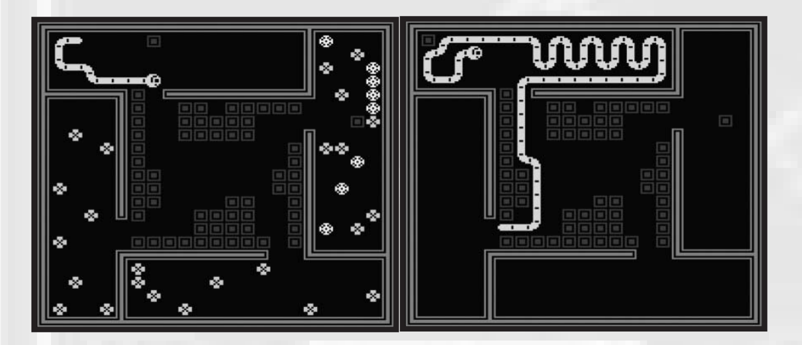
¡Socorro! Este nivel asusta nada más aparecer (1), ya que en el centro aparece un cuadrado lleno de bloques azules móviles que deberemos atravesar para poder acceder a las cuatro zonas que contienen las plantas. Lo primero que haremos, antes de comernos nada, es abrir los caminos hacia estas zonas para poder acceder a ellas sin problemas (2).





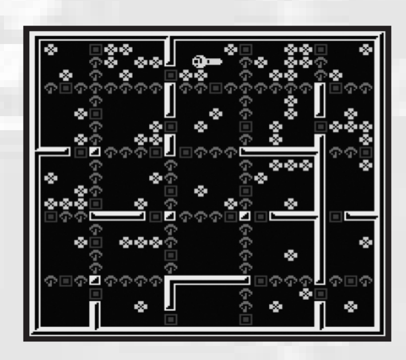
Una vez hecho esto el nivel es muy sencillo (3), ya que nos basta con entrar en las cuatro áreas en el orden de las agujas del reloj para ir comiendo los tréboles y las flores que van a ir apareciendo (4). En este nivel, como comentábamos

al principio, no aparecen setas rojas.



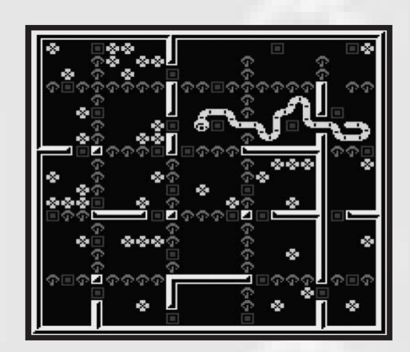
Nivel 7

Y llegamos al nivel, en mi opinión, más complicado de todo el juego, ya que nos encontramos con un laberinto formado por 25 pequeños habitáculos separados por paredes o setas (1). A base de muchos intentos encontré un camino para poder terminarlo con éxito, puede que no sea el mejor, pero servirá para poder avanzar en el juego.

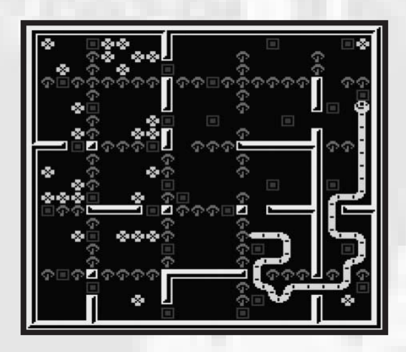


En este nivel tampoco aparecen setas al comernos las flores blancas que irán apareciendo, aunque el nivel ya es lo suficientemente complicado como para echar las campanas al vuelo.

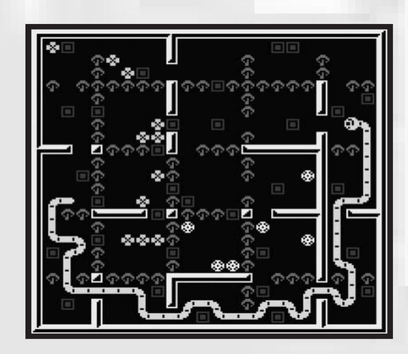
Lo primero que haremos será ir al habitáculo de nuestra derecha y seguir este camino de habitáculos: derecha, abajo, izquierda e izquierda, llegando a la situación que se puede ver en la imagen (2). ¡Cuidado con ser ambiciosos! No hay que intentar comer todas las plantas, fijaos bien en el trébol que queda en el habitáculo superior derecho. También hay que tener cuidado para no mover demasiado los bloques azules, ya que podemos encerrarnos muy fácilmente.



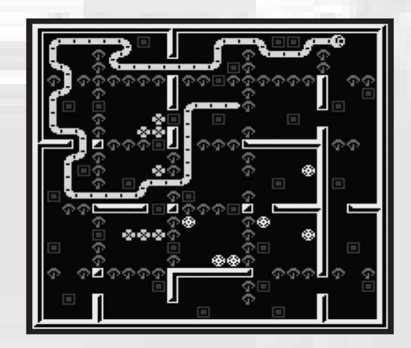
El segundo paso será realizar el siguiente recorrido: abajo, izquierda, abajo, abajo, derecha, y tres para arriba, llegando a la situación que encontramos en (3).



¡Ahora sí! Estamos en situación de comernos el trébol que nos dejamos antes haciendo el tercer recorrido que nos llevará por los habitáculos más externos: arriba, cuatro a la izquierda, cuatro hacia abajo, cuatro a la derecha y tres para arriba. Nuevamente nos deberemos dejar algunos tréboles por el camino para no quedarnos encerrados (4).

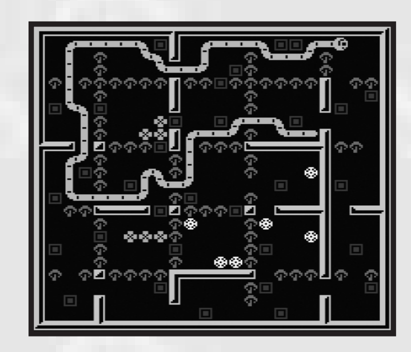


El siguiente recorrido será: dos a la izquierda, uno abajo, dos a la izquierda, dos hacia arriba y cuatro a la derecha (5). Nos dejamos un trébol por el camino pero nos comemos los que nos habíamos dejado anteriormente.

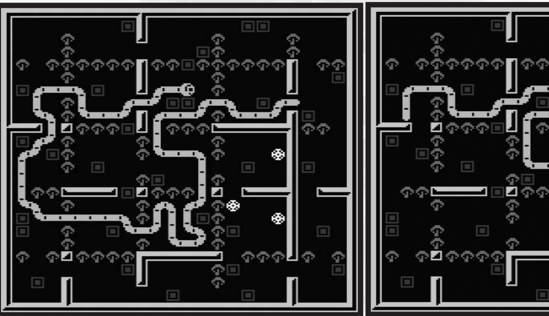


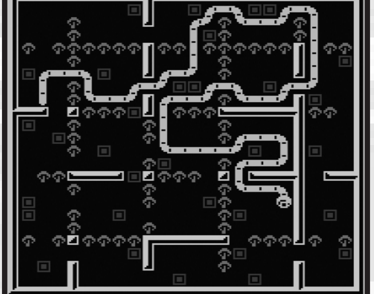
Ahora seguiremos durante un rato la cola de la serpiente para comernos el trébol que nos hemos dejado y volver a la

misma situación que tenemos: abajo, dos a la izquierda, abajo, dos a la izquierda, dos arriba y cuatro a la derecha (6).



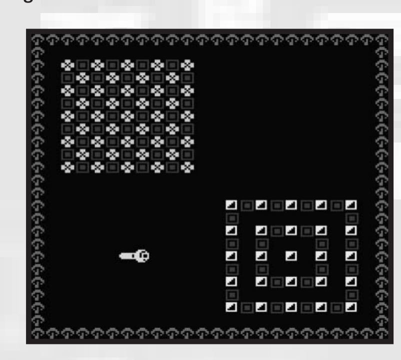
El siguiente recorrido nos llevará por los habitáculos que aún tienen tréboles: uno abajo, dos a la izquierda, dos abajo, dos a la izquierda, dos arriba y dos a la derecha (7). Ya sólo nos quedan dos habitáculos más con flores blancas para terminarnos el nivel: arriba, dos a la derecha, abajo, dos a la izquierda, abajo, derecha y abajo (8).



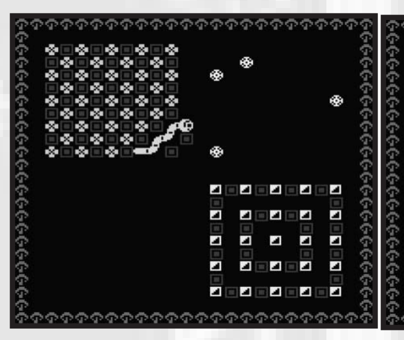


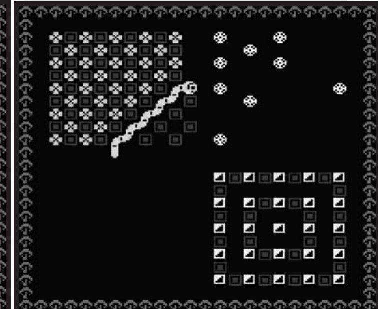
Nivel 8

Tras la complejidad del anterior nivel, nos encontramos con uno muy sencillo aunque de apariencia complicado (1). Al principio vemos un cuadrado formado por tréboles y cuadrados azules y otro por paredes y cuadrados azules. Bien, este último está ahí para despistar, ya que no tendremos que tocarlo en ningún momento.

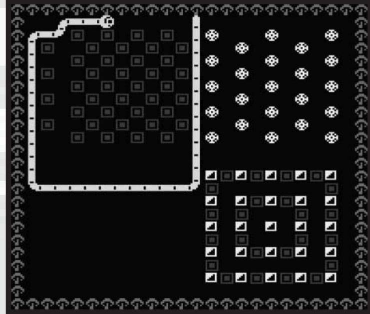


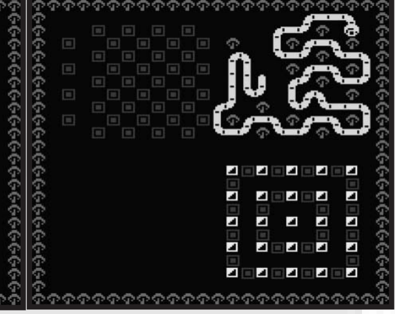
Comenzaremos comiéndonos los tréboles del cuadrado, moviendo todos los bloques azules hacia la derecha. Comeremos los tréboles en diagonales sucesivas comenzando por la más baja y subiendo: (2 y 3).





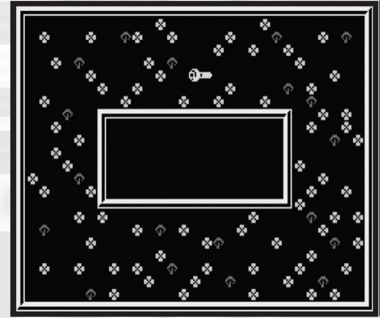
Cuando hayamos comido todos los tréboles, nos habrán aparecido veintitrés flores blancas (4), que nos comeremos con cuidado ya que entre ellas aparecen nuevamente setas rojas (5).

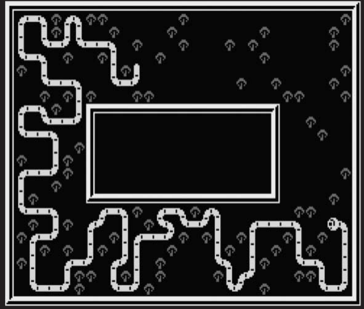




Nivel 9

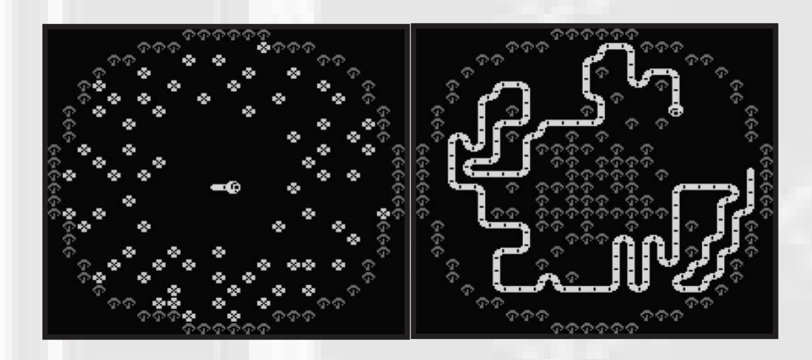
Contrariamente a lo esperado, los niveles que quedan son muy fáciles. El nivel nueve es una muestra de ello (1). Aquí basta aplicar la estrategia general de ir despacito y con tranquilidad hasta comernos todos los tréboles y las flores blancas que aparecen, esquivando tanto las setas que estaban originalmente como las que aparecen durante el juego (2).





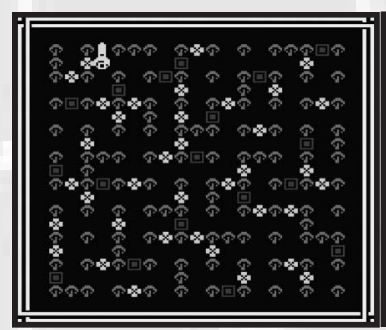
Nivel 10

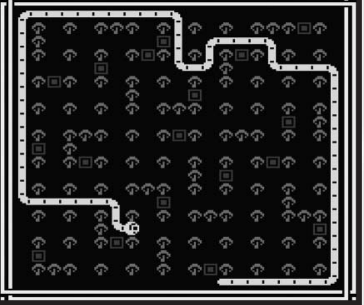
Otro nivel sencillito (1) que resolveremos fácilmente siguiendo la misma estrategia que en el anterior (2).



Nivel 11

Y un nivel más bastante sencillo (1). Aquí deberemos tener cuidado de no mover los bloques azules, ya que están estratégicamente situados para lograr que acabemos encerrándonos. En este nivel tampoco aparecen nuevas setas, por lo que terminarlo será un juego de niños (2).





¿Ya está?

Pues sí, ya hemos terminado el juego. Personalmente hubiera preferido un final a que se repitieran indefinidamente los niveles, ya que una vez que los sabemos resolver el juego pierde todo su aliciente. El grado de dificultad es correcto hasta el nivel 8 (se agradece un nivel sencillo tras la excesiva complejidad del séptimo), pero los siguientes tres niveles son extremadamente simples y esto hace que el juego pierda su aliciente en la recta final.

Comentarios finales

Desde el punto de vista gráfico, este juego está realizado de manera impecable para el año 1986. El uso del color es escaso, pero correcto. El movimiento de la serpiente está muy bien conseguido y los objetos móviles (corazones, bonus y vidas extra) se ven perfectamente y no hay ningún parpadeo de sprites en ningún momento.

El juego está realizado en Screen 1, lo cual explica el límite de colores por patrón, aunque permite una rápida actualización de la tabla de patrones para simular el movimiento de la serpiente. El uso de sprites es más que correcto, ya que todos los objetos (serpiente incluída) están formados por una mezcla de sprites y patrones.

El apartado sonoro está algo descompensado. Sólo hay una melodía, bastante pegadiza, por cierto, en los títulos de crédito, lo cual choca con la voz de Ronald van der Putten digitalizada que nos da la bienvenida al juego (WELCOME TO SNAKE IT), nos indica que pasamos al siguiente nivel (NEXT SCREEN) y nos anuncia el fin del juego (GAME OVER). También podremos escuchar unos soniditos cuando los corazones, bonus y vidas extra estén pululando por la pantalla, aunque no hay ninguna música mientras jugamos.

Quizá el fallo más gordo del juego sea que, como hemos visto, los niveles posteriores al siete son bastante más sencillos en comparación con éste. ¿Acaso no se podía haber cuidado un poco más la jugabilidad sacrificando otros aspectos como pudiera ser la voz digitalizada? En mi opinión el crecimiento de la dificultad es correcto hasta ese nivel (y si me apuráis hasta el siguiente), para caer drásticamente en los tres últimos niveles del juego.

Como dato curioso, si le echamos un vistazo a la ROM en un editor hexadecimal, vemos que dentro hay un mensaje dedicado a los posibles hackers: "Sun is going down, disappearing behind the boulevard. It's Sharky's night tonight!".

En conclusión estamos ante un juego muy divertido y que puede tenernos entretenidos una buena tarde de lluvia (de hecho a mí me tuvo muchas). Analizándolo desde el punto de vista de hoy en día hay muchas cosas mejorables, las cuales lanzo aquí por si alguien se anima a recoger el guante y hacer una versión actualizada del juego:

-Mejoras gráficas. ¿Por qué no utilizar Screen2 con patrones más coloridos? En lugar de redefinir los patrones se puede utilizar la tabla de nombres para simular el movimiento de la serpiente.

-Mejoras sonoras. Quizá prescindir de las voces digitalizadas y ambientar el juego con alguna musiquilla que otra.

-Más niveles. Este es el punto más flaco del juego: se hace corto. Se necesitan más niveles, más retos. ¿Por qué no hacer niveles más grandes y permitir que éstos scrollen? Más objetos que puedan influir en el juego permitiría que los sucesivos niveles fueran complicándose más.

Espero que alguien se anime y pronto podamos disfrutar de "Snake It 2, return of the Snake!"

Sapphire

Hazte tu cable **RGB** para el ***One Chip MSX***

Por **Manuel Pazos**

El 1chip MSX dispone de 3 salidas de vídeo: compuesto, S-Video y VGA. Podemos seleccionar mediante el DIP Switch 1 y 2 cuál queremos activar:

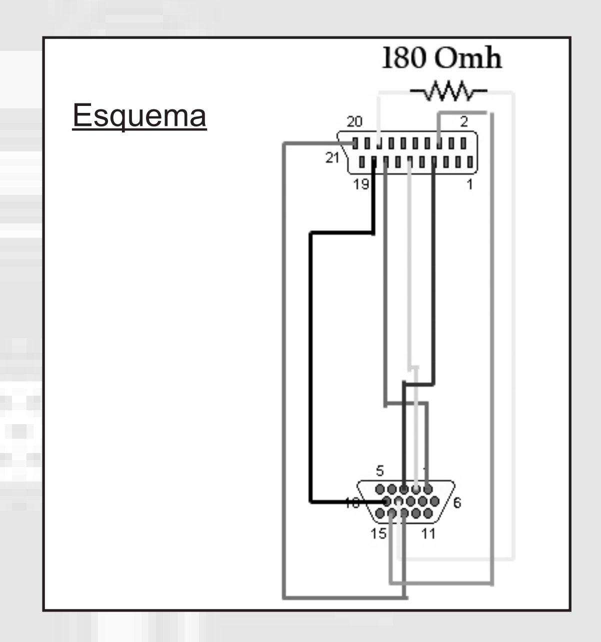
Salida	Switch 1 - 2
Video compuesto y S-Video	OFF-OFF
RGB (15kHz)	OFF-ON
VGA (31kHz)	ON-OFF
VGA (31kHz) Brillo fuerte	ON-ON

Material necesario:

-Soldador
-Estaño
-Conector VGA15 macho (15 PIN HIGH-DENSITY D-SUB MALE)
-Conector Euroconector macho (21 PIN SCART MALE)
-Cable apantallado de 8 hilos
-Conector RCA macho blanco (opcional)

Las señales de video y audio para el modo RGB se obtienen del conector VGA. El único problema es que el audio suministrado por este conector, no está mezclado con la señal SOUND-IN del slot de cartuchos, por lo que no se oirá ninguna fuente de sonido externa como un SCC, un Music Module, etc. Sin embargo, si el sonido SCC lo genera el propio OCM, no habrá problema (por ejemplo activando el funcionamiento del slot 2 como SCC).

La solución a este pequeño inconveniente es coger la señal de audio del jack blanco en vez del conector VGA.



Señales

Señal	VGA	Euroconector
R	Red Video	RGB Red In
G	Green Video	RGB Green In
B	Blue Video	RGB Blue In
Composite SYNC (15kHz)	HSYNC	Composite Video In
Audio	VSYNC	Audio In Left (Mono)
GND	Sync Ground	Composite Video Ground
BLNK	+5V	Blanking signal

Pines

VGA	Euroconector
1	15
2	11
3	7
13	20
14	6
10	17
9	16


Quality MSX software
Best price
Cartridge
MajiKazo
traffic jam
IMANOK
© 2006-2008 IMANOK
MSX
CAOS BEGINS
Hikaru Games 2007
An Epic Adventure by Hikaru Games
ROM
User Manual
The MSX Cartridge Shop
http://www.msxcartridgesoft.com